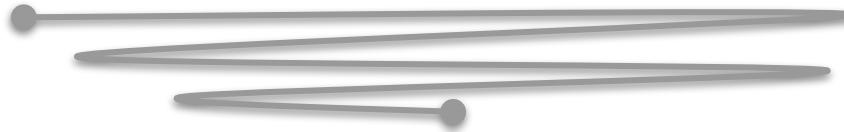
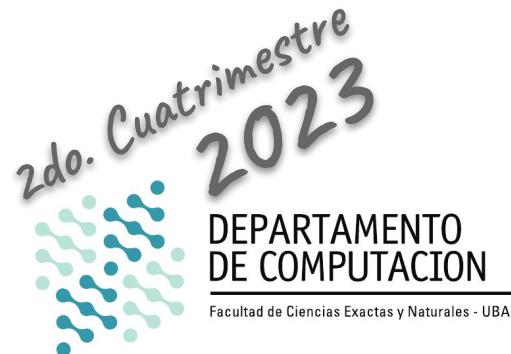


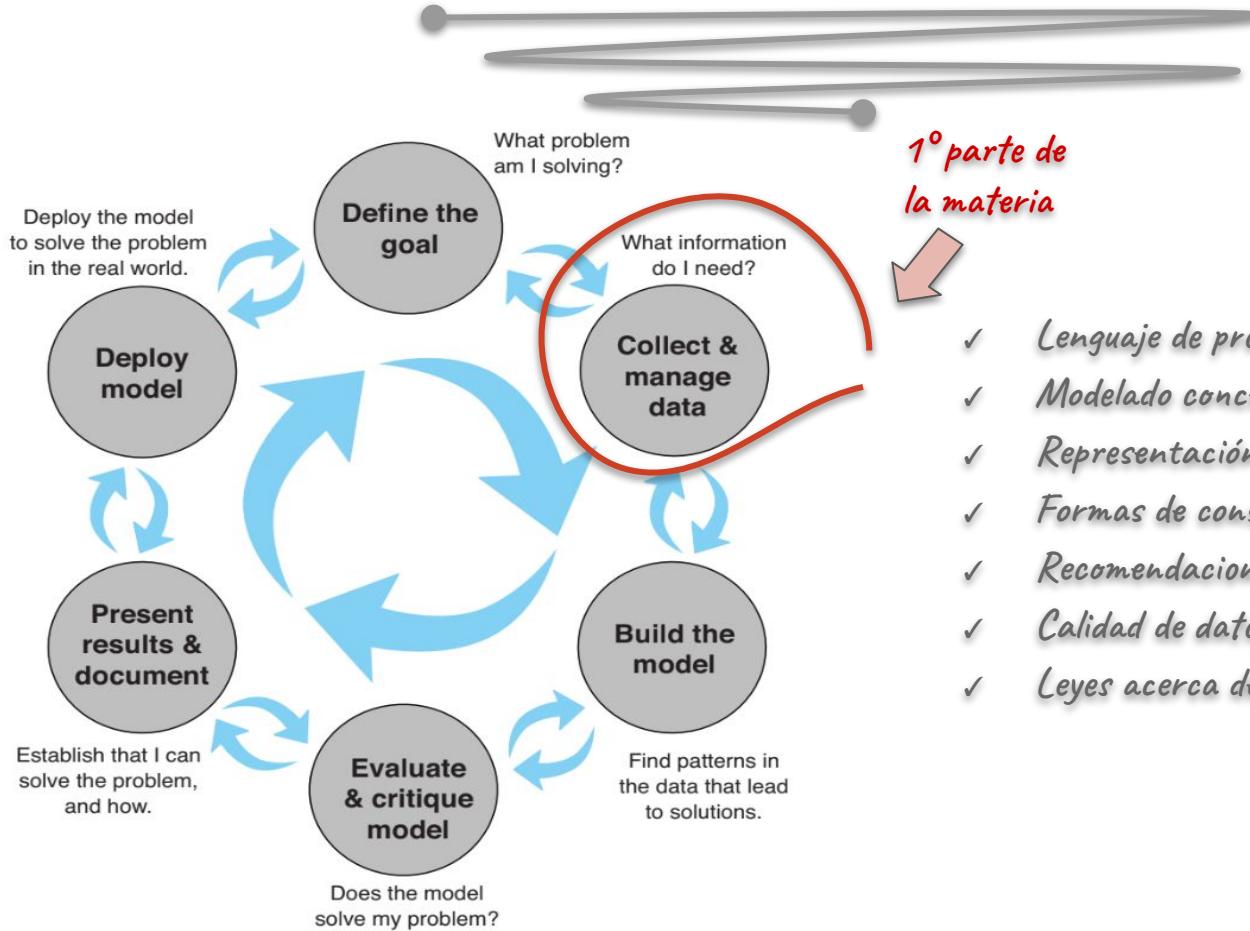
# *Laboratorio de Datos*



## *Aprendizaje No Supervisado*



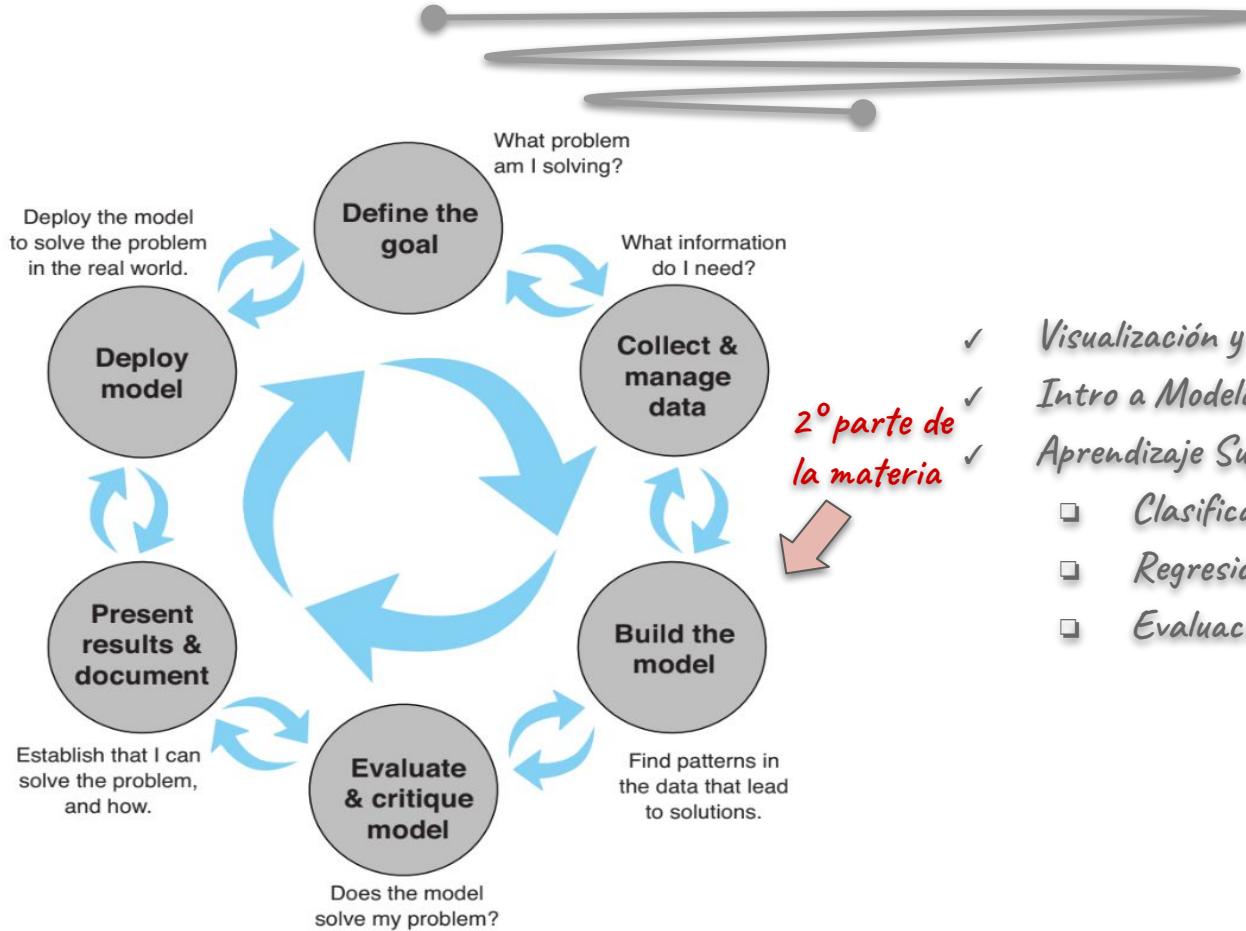
# Recorrido de la materia (hasta ahora)



1º parte de  
la materia

- ✓ Lenguaje de programación (Python)
- ✓ Modelado conceptual de los datos (DER)
- ✓ Representación de los datos (modelo relacional)
- ✓ Formas de consultar los datos (AR/SQL)
- ✓ Recomendaciones para el diseño (Normalización)
- ✓ Calidad de datos
- ✓ Leyes acerca de la Protección de Datos

# Recorrido de la materia (hasta ahora)



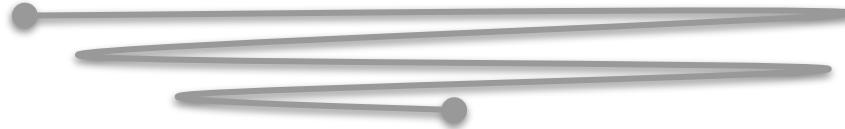
- ✓ Visualización y Exploración de los datos
- ✓ Intro a Modelado: Clasificación y Regresión
- ✓ Aprendizaje Supervisado
  - Clasificación: Árboles de decisión, KNN
  - Regresión: Regresión Lineal, KNN
  - Evaluación

# Laboratorio de Datos

## Aprendizaje no supervisado

... por Manuela Cerdeiro (y modificaciones de P. Turjanski)

*Actividad grupal*



# Consigna



## Consigna

- Conformar equipos de 3 estudiantes
- Agrupar a los 9 superhéroes en grupos  
(generar más de 1 grupo y menos de 9, justificando)
- Tienen 5 minutos
- Al finalizar, subir una foto de los grupos al siguiente link: [Carpeta Fotos](#)

Iron Man	Cyborg	Warlock
Thor	Superman	Tormenta
Ant-Man	Batman	Guepardo

Una forma ...

ROBOTS



VOLADORES



ANIMALES



Otra forma ...

AVENGERS

LIGA DE LA JUSTICIA

X-MEN



Otra, otra forma ...

DC COMICS

Batman



Superman



Cyborg



MARVEL

Thor



Iron Man



Ant-Man



Warlock



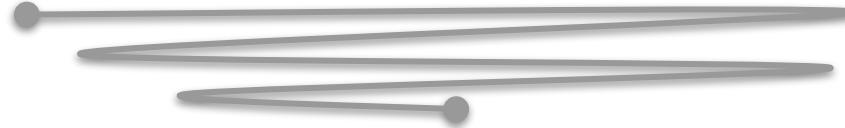
Tormenta



Guepardo



## Consigna



¿Qué tipos de clasificaciones tuvimos?

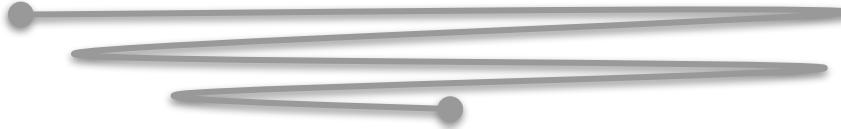
- por características de cada personaje
- por grupo de pertenencia
- por editorial

La selección de atributos nos condiciona el agrupamiento.



¿Cuál es el agrupamiento correcto?

*Aprendizaje*



*Aprendizaje [Si/NO] Supervisado*

# Aprendizaje Supervisado

## 1. Aprendizaje Supervisado.

Hasta ahora (Árboles de decisión, Regresión, KNN, etc.), las observaciones de entrenamiento tenían variables predictoras y un valor a predecir conocido (etiquetas)

Ejemplo:



The diagram illustrates the process of supervised learning. It shows two tables: 'Entrenamiento' (Training) on the left and 'Predicción' (Prediction) on the right. A curved arrow points from the 'Entrenamiento' table to the 'Predicción' table, indicating that the training data is used to make predictions for new data.

Entrenamiento			Predicción		
	RU	DI		RU	DI
1	0	104.00	1	25	
2	50	106.00	2	600	
3	100	112.30	3	1500	
4	200	117.00			
5					

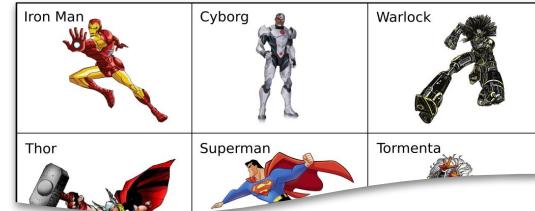
# Aprendizaje No Supervisado

## 2. Aprendizaje NO Supervisado.

No conocemos a priori la etiqueta

- Se trata de inferir modelos sobre datos de los que se desconoce la salida deseada
- Las técnicas tratan de identificar características similares entre los datos para proponer una clasificación correcta, detectando patrones que quizás no se detectan a simple vista
- Existen diferentes tipos de algoritmos, nosotros nos vamos a centrar en algoritmos de clustering

Ejemplo 1



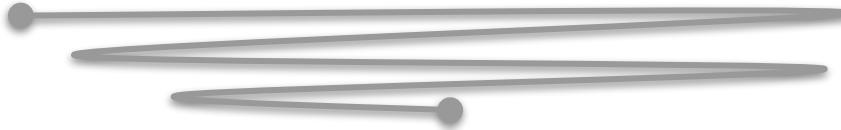
DC COMICS



MARVEL



# *Aprendizaje No Supervisado*



*Clustering*

# Aprendizaje No Supervisado - Clustering

Objetivo. Agrupar los datos en función de sus características

- Se basa en que ...
  - los datos que pertenecen a un mismo grupo tienen muchas similitudes entre sí
  - los de distintos grupos se parecen poco



C1

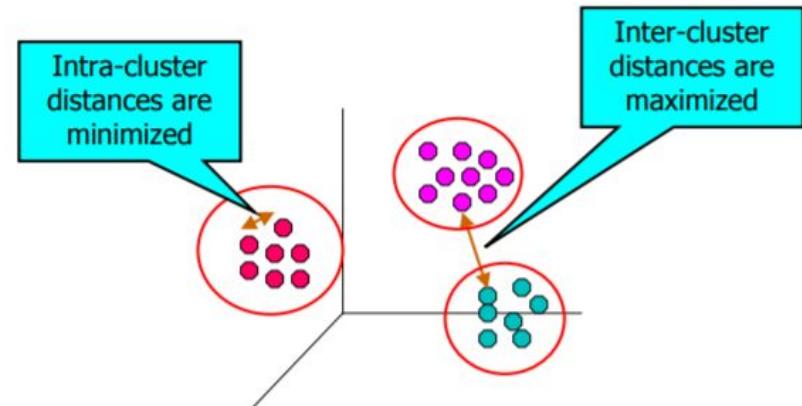
C2

# Aprendizaje No Supervisado - Clustering

Objetivo. Encontrar grupos de instancias (clusters) a partir de información en los datos que describan objetos y sus relaciones.

Instancias de un cluster tienen que ser:

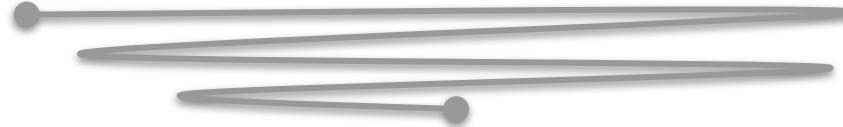
- similares entre sí y
- diferentes a las de otros clusters



Tan, Steinbach & Kumar, Introduction to Data Mining

[https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8\\_basic\\_cluster\\_analysis.pdf](https://www-users.cs.umn.edu/~kumar001/dmbook/dmslides/chap8_basic_cluster_analysis.pdf)

# Aprendizaje No Supervisado - Clustering

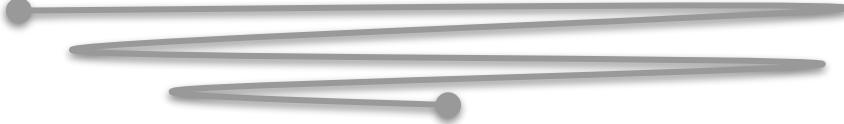


## Ejemplos.

- Agrupar en películas en Buenas/Malas
- Análisis de redes sociales
- Segmentación de clientes
- Clasificación de especies (seres vivos)
- Análisis de enfermedades



# Aprendizaje No Supervisado - Clustering



## Algoritmos

- KMeans
- DBScan
- Agrupación jerárquica
- etc.

# Aprendizaje No Supervisado - Clustering

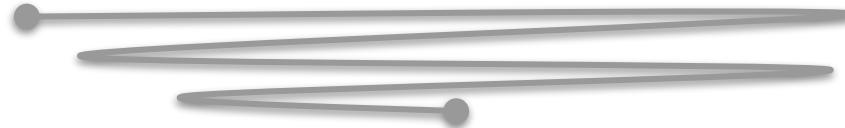


*K-Means*

# K-MEANS CLUSTERING 5 PASOS!



## *Algoritmo K-Medias (K-Means)*



- Algoritmo

Es un método iterativo:

1. **Inicialización:** se elige la localización de los centroides de los K grupos aleatoriamente y se asigna cada dato al centroide más cercano
2. **Actualización:** se actualiza la posición del centroide a la media aritmética de las posiciones de los datos asignados al grupo
3. **Asignación:** se asigna cada dato al centroide más cercano

Repite 2 y 3 hasta que la asignación es estable o se agotan las iteraciones permitidas.

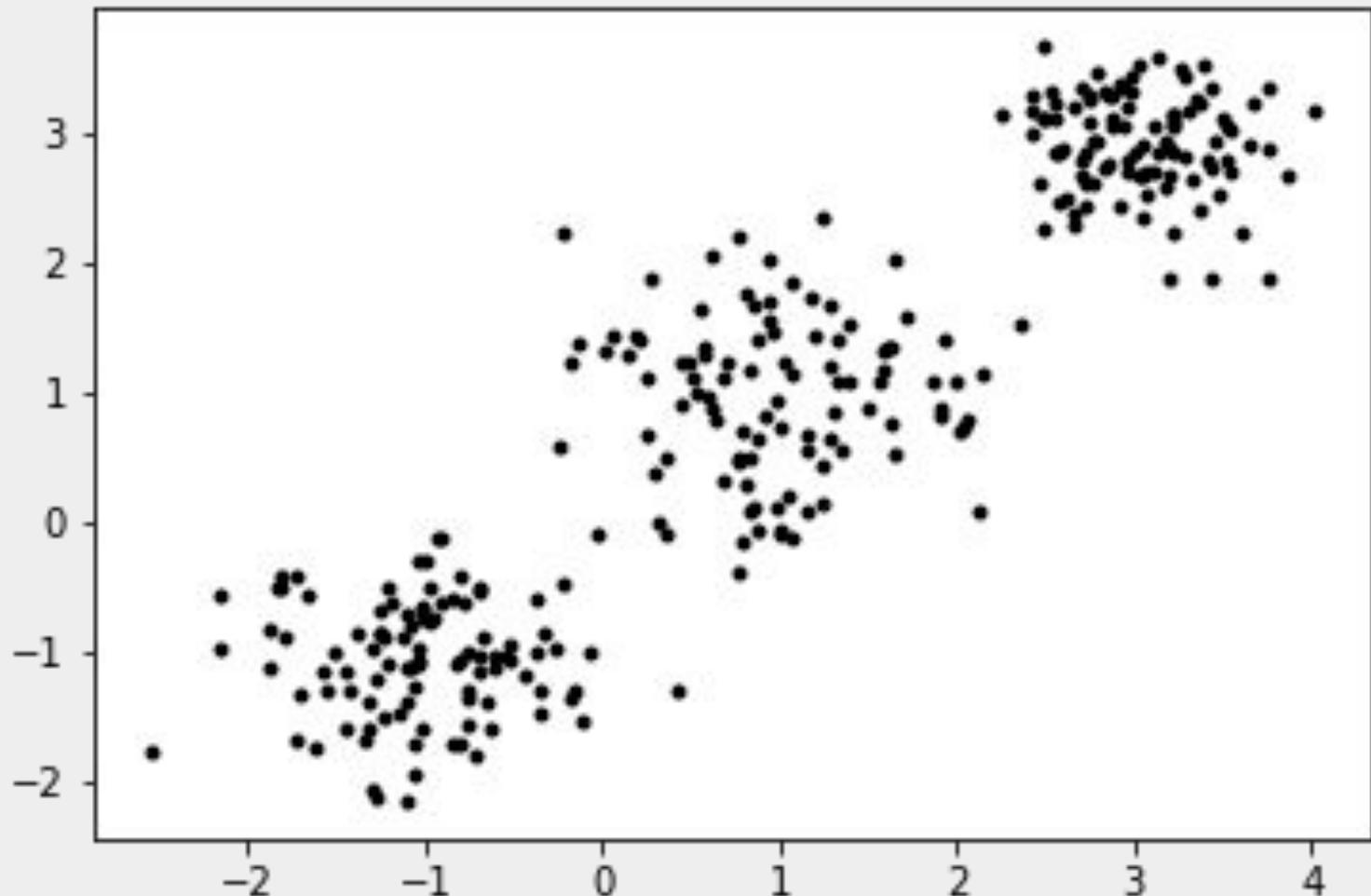
- Es un método, simple, rápido y generalmente efectivo para detectar clusters

*Datos de entrada*

Ejecución ( )

*Datos de entrada*

Ejecución (Dataset)

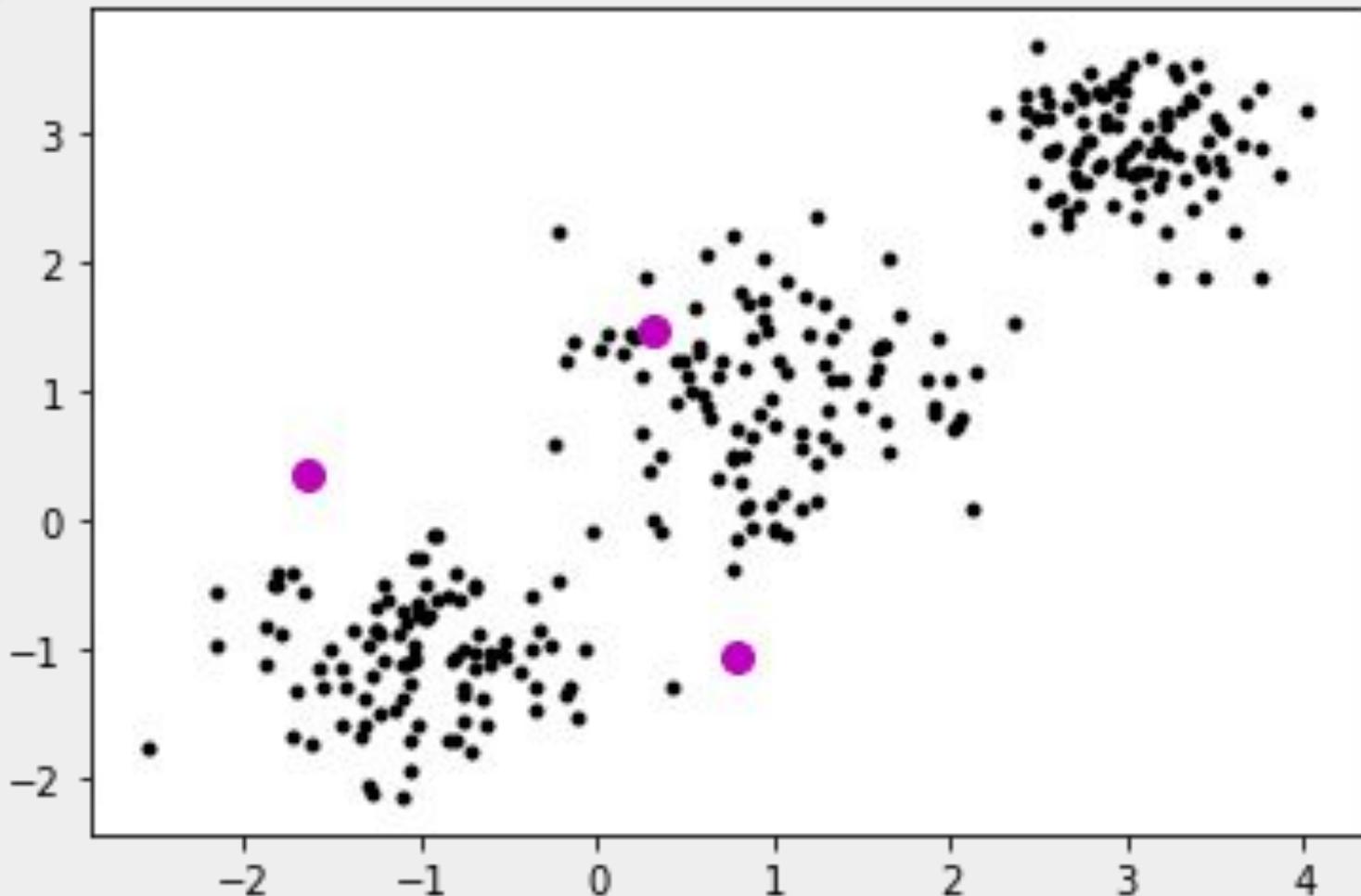


Usamos  $k = 3$

Sorteamos los centroides

Ejecución (Dataset)

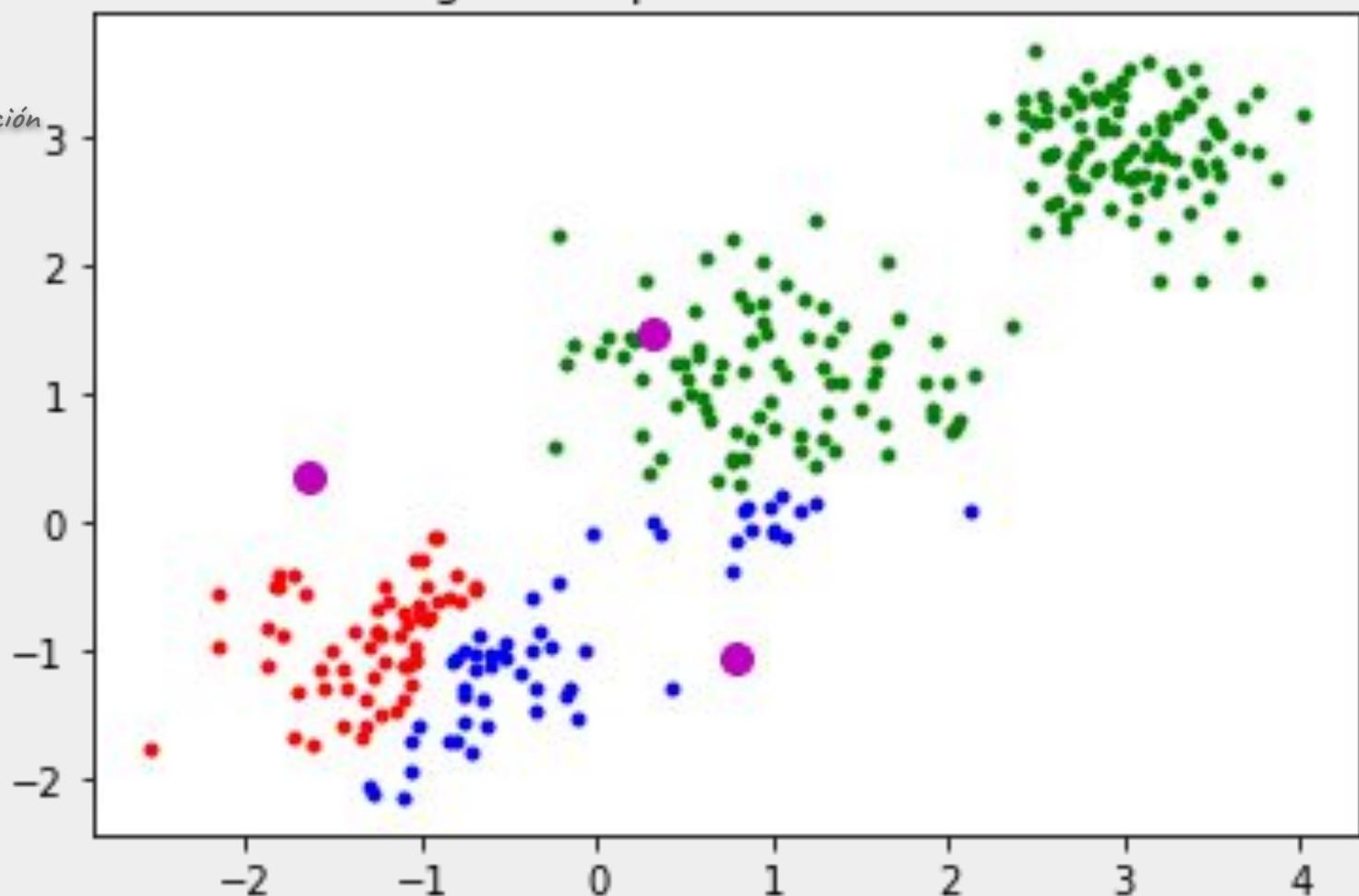
1. Inicialización



## Asignamos puntos a centroides

Ejecución (Dataset)

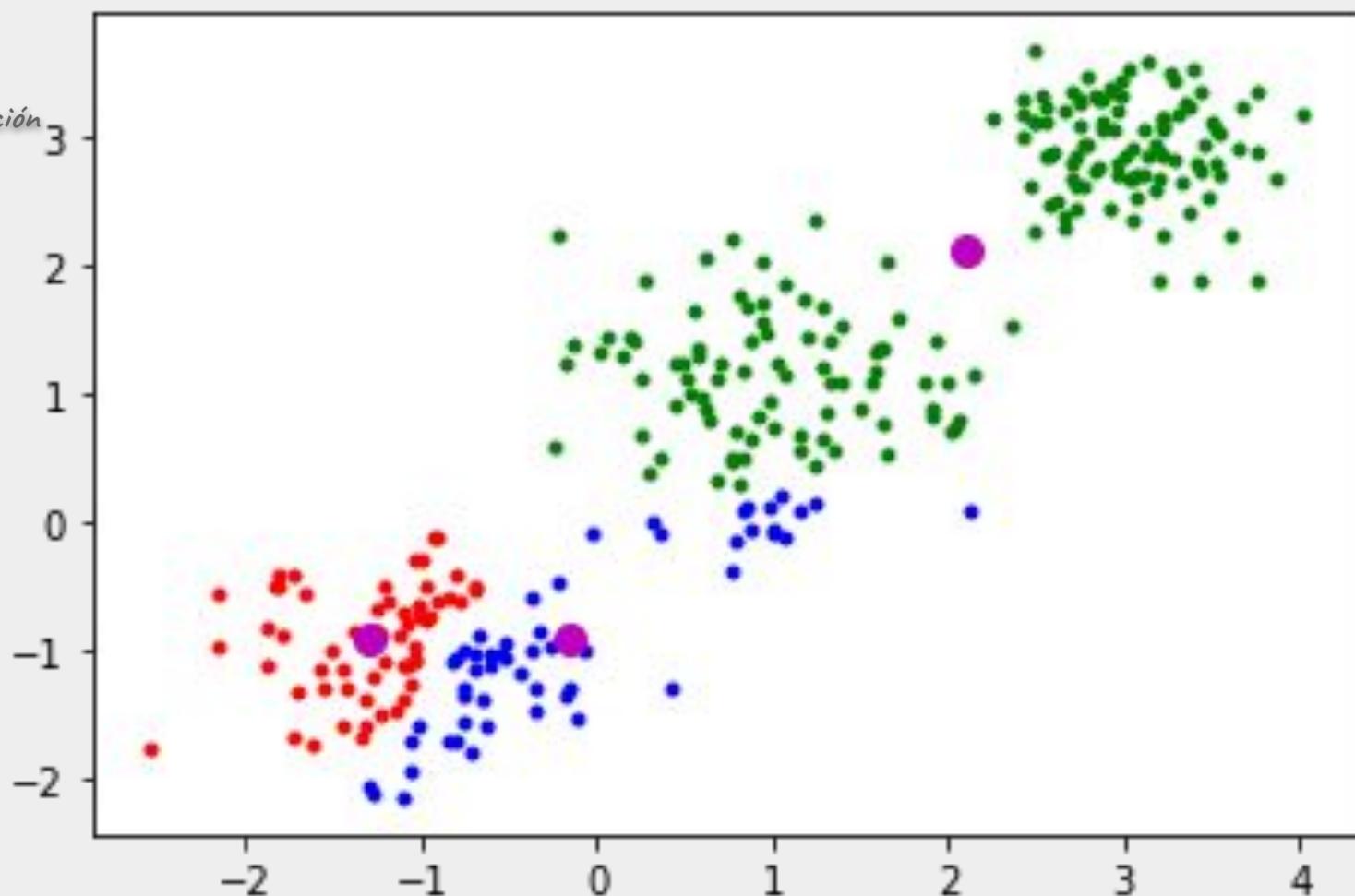
1. Inicialización y Asignación



## Reubicamos centroides

Ejecución (Dataset)

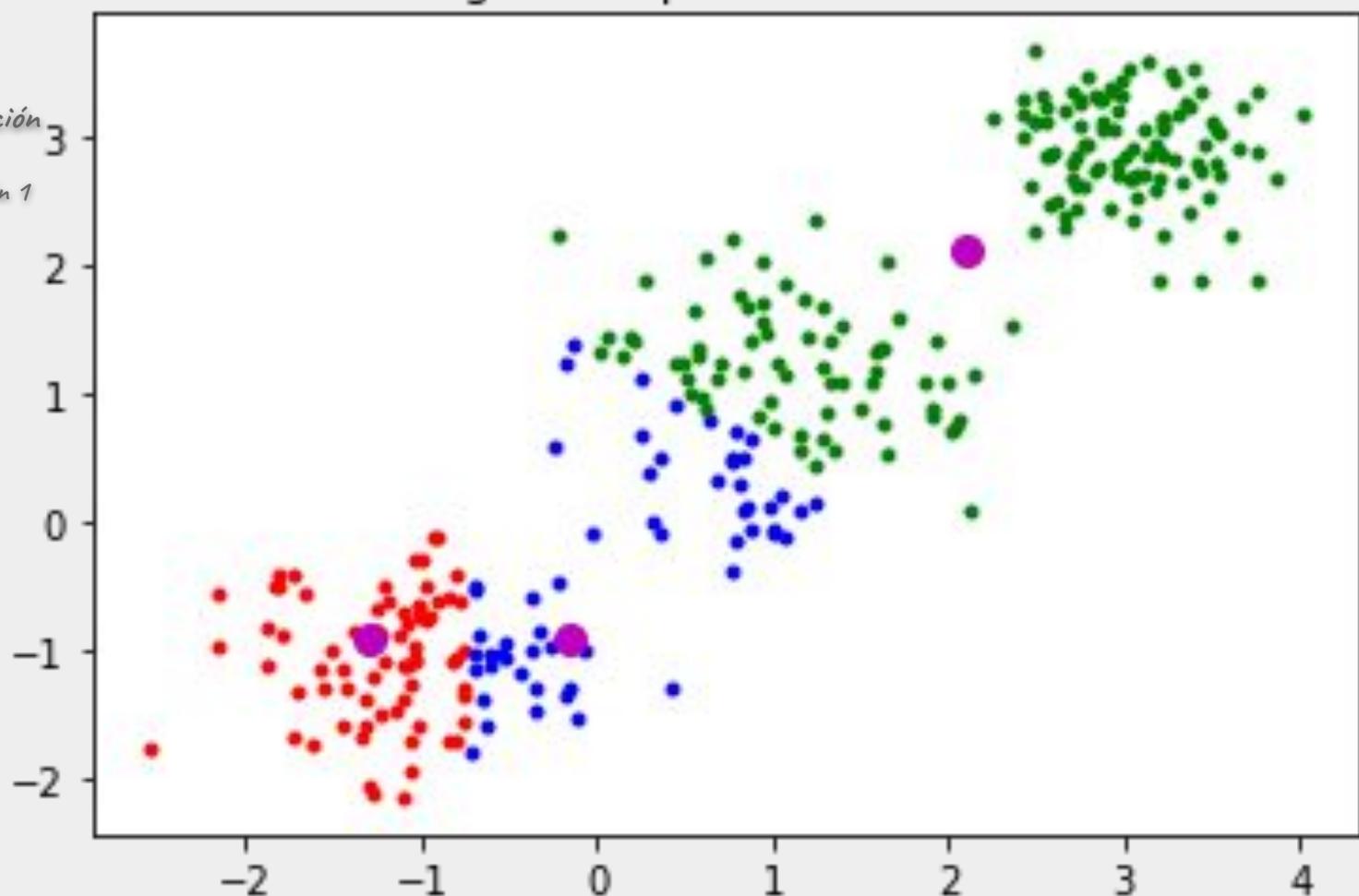
1. Inicialización y Asignación
2. Actualización



## Reasignamos puntos a centroides

Ejecución (Dataset)

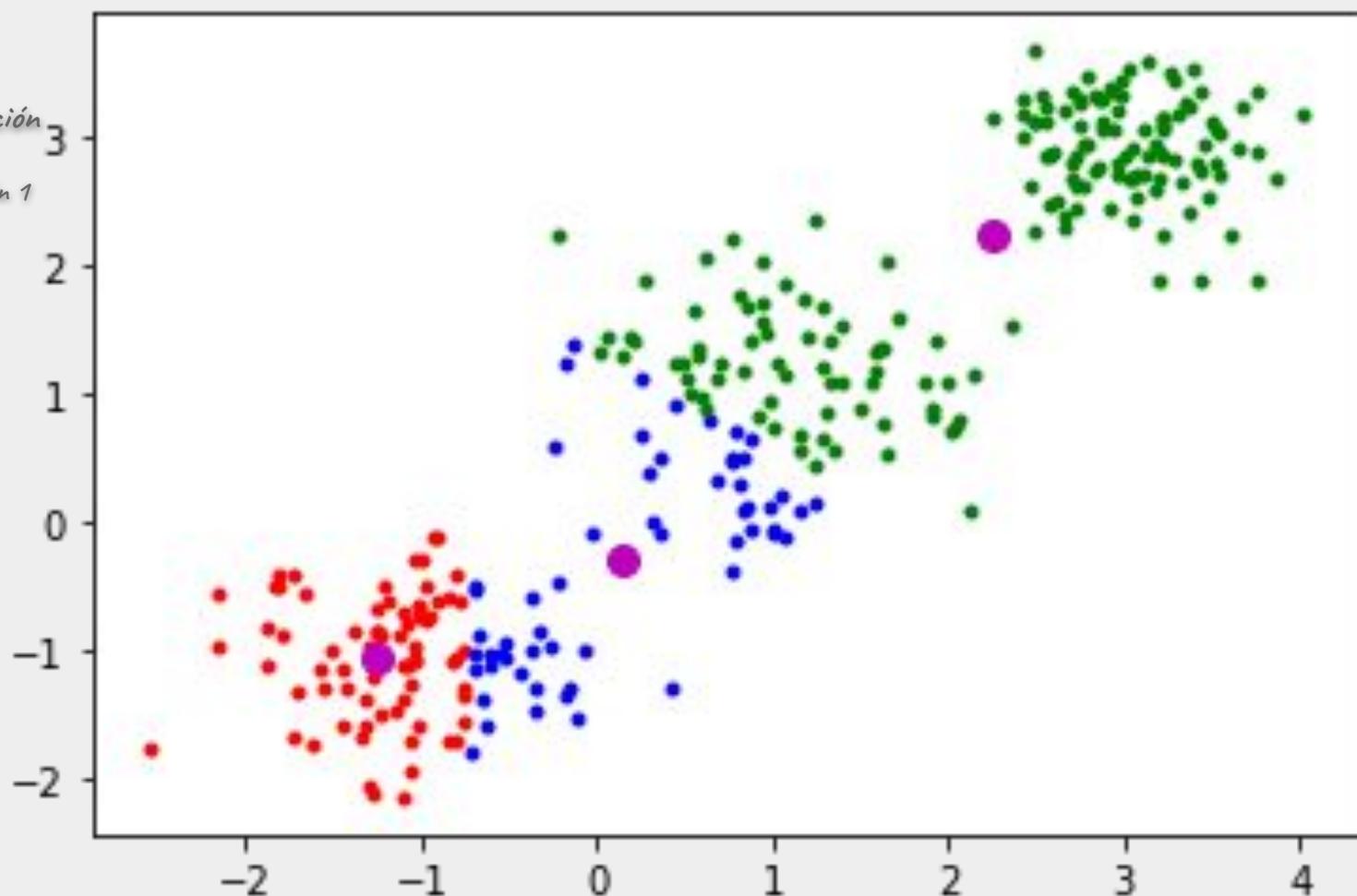
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación



## Reubicamos centroides

Ejecución (Dataset)

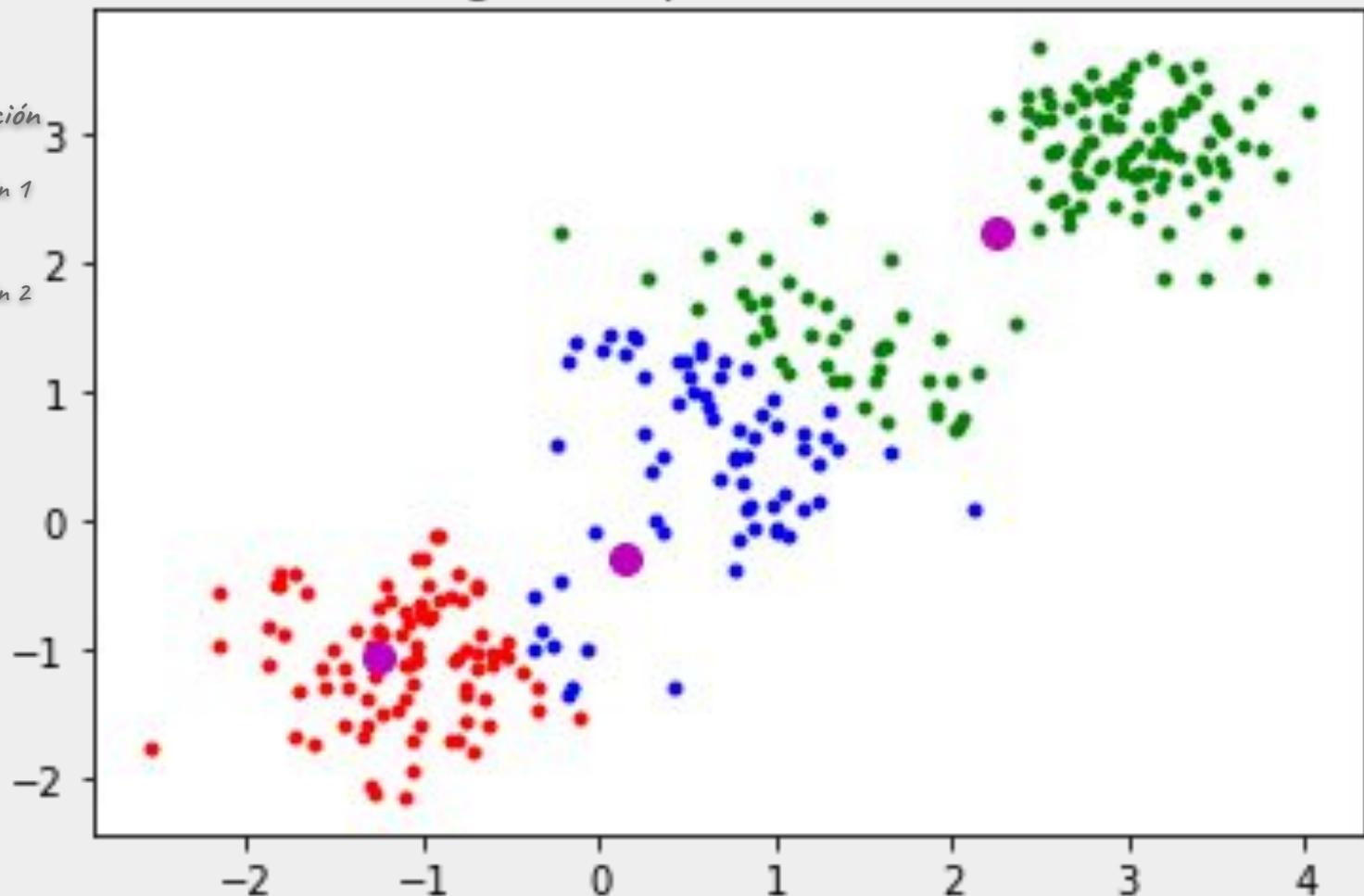
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización



## Reasignamos puntos a centroides

Ejecución (Dataset)

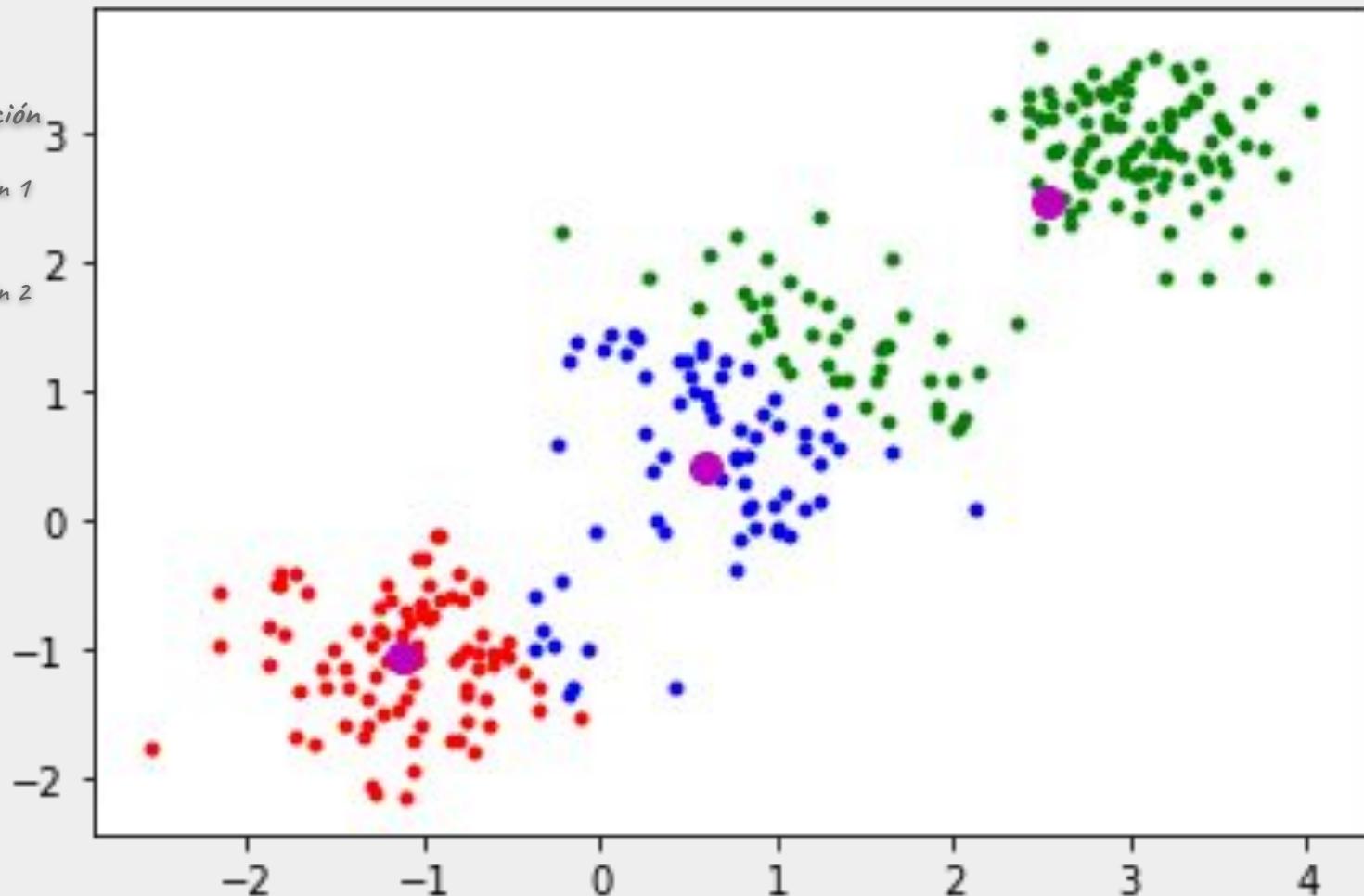
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación



## Reubicamos centroides

Ejecución (Dataset)

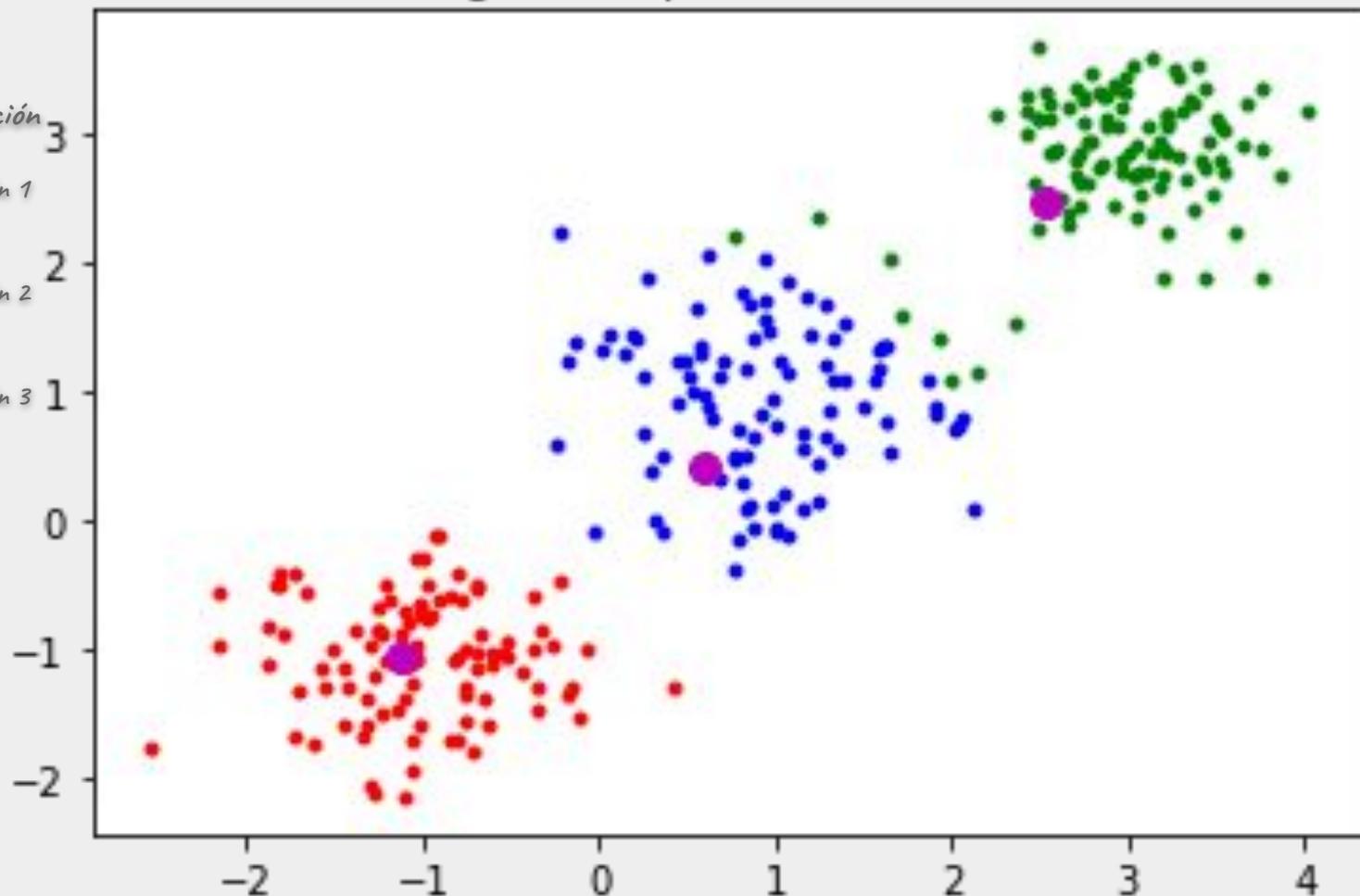
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización



## Reasignamos puntos a centroides

### Ejecución (Dataset)

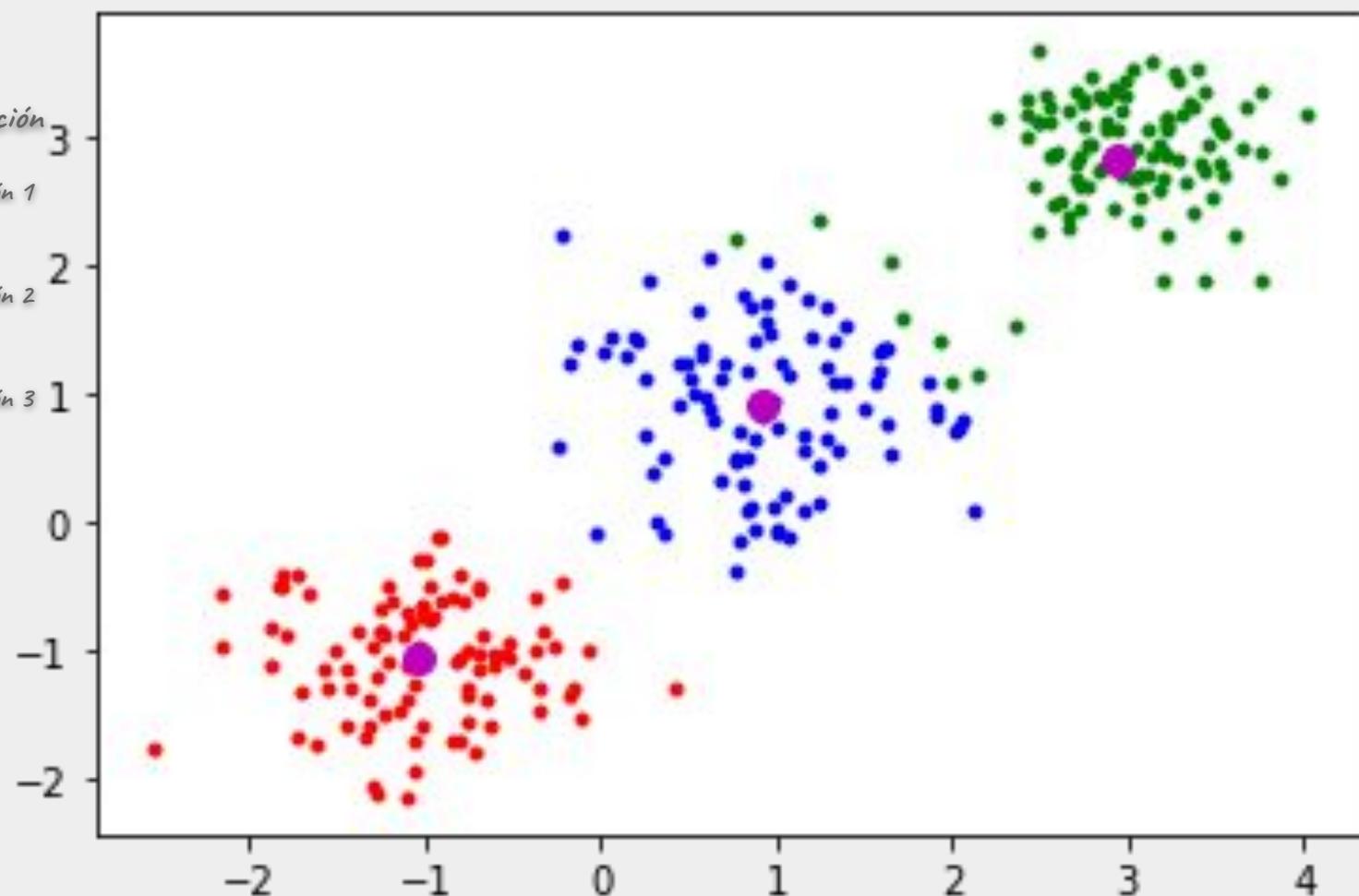
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación



## Reubicamos centroides

Ejecución (Dataset)

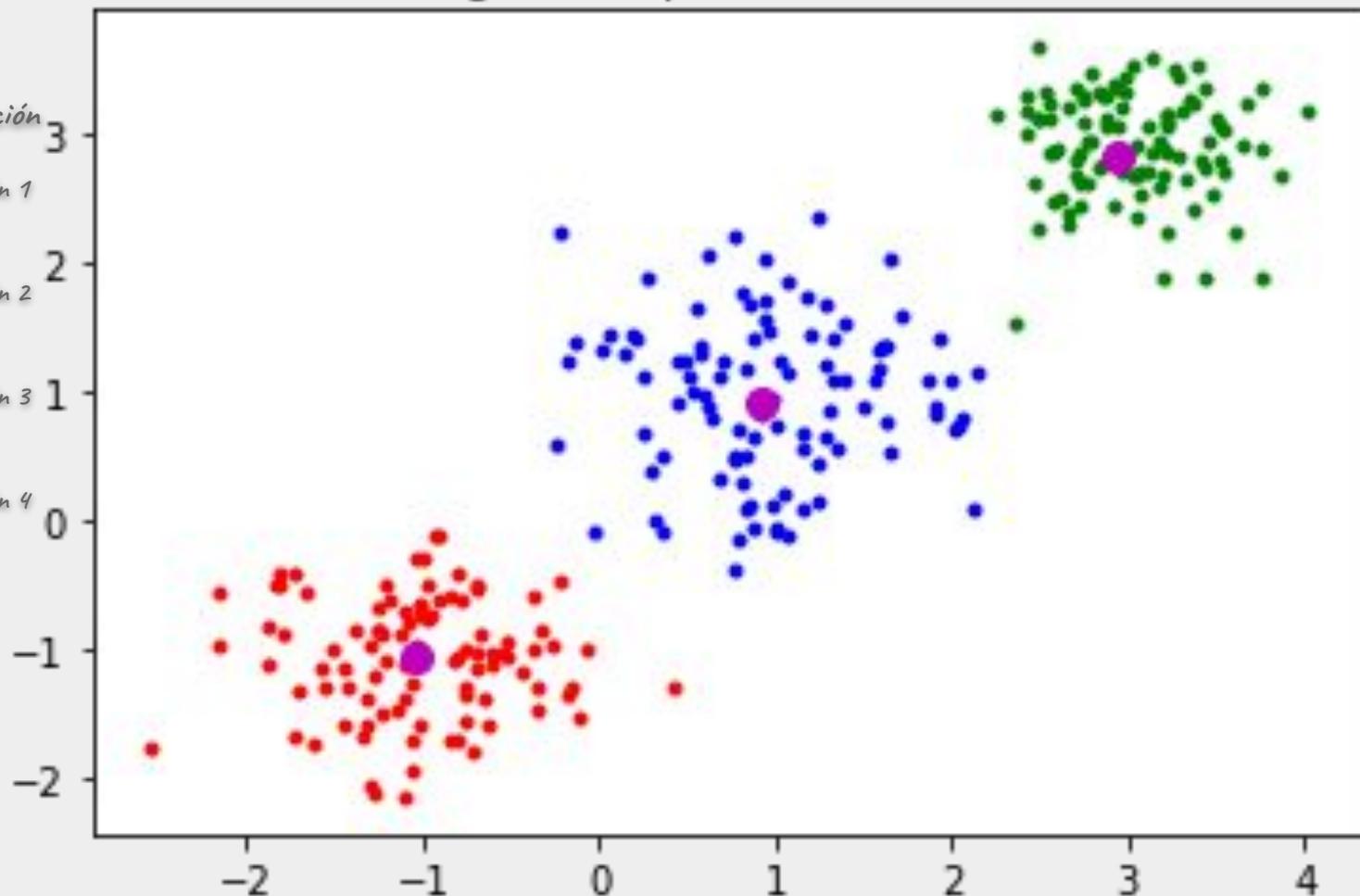
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización



## Reasignamos puntos a centroides

### Ejecución (Dataset)

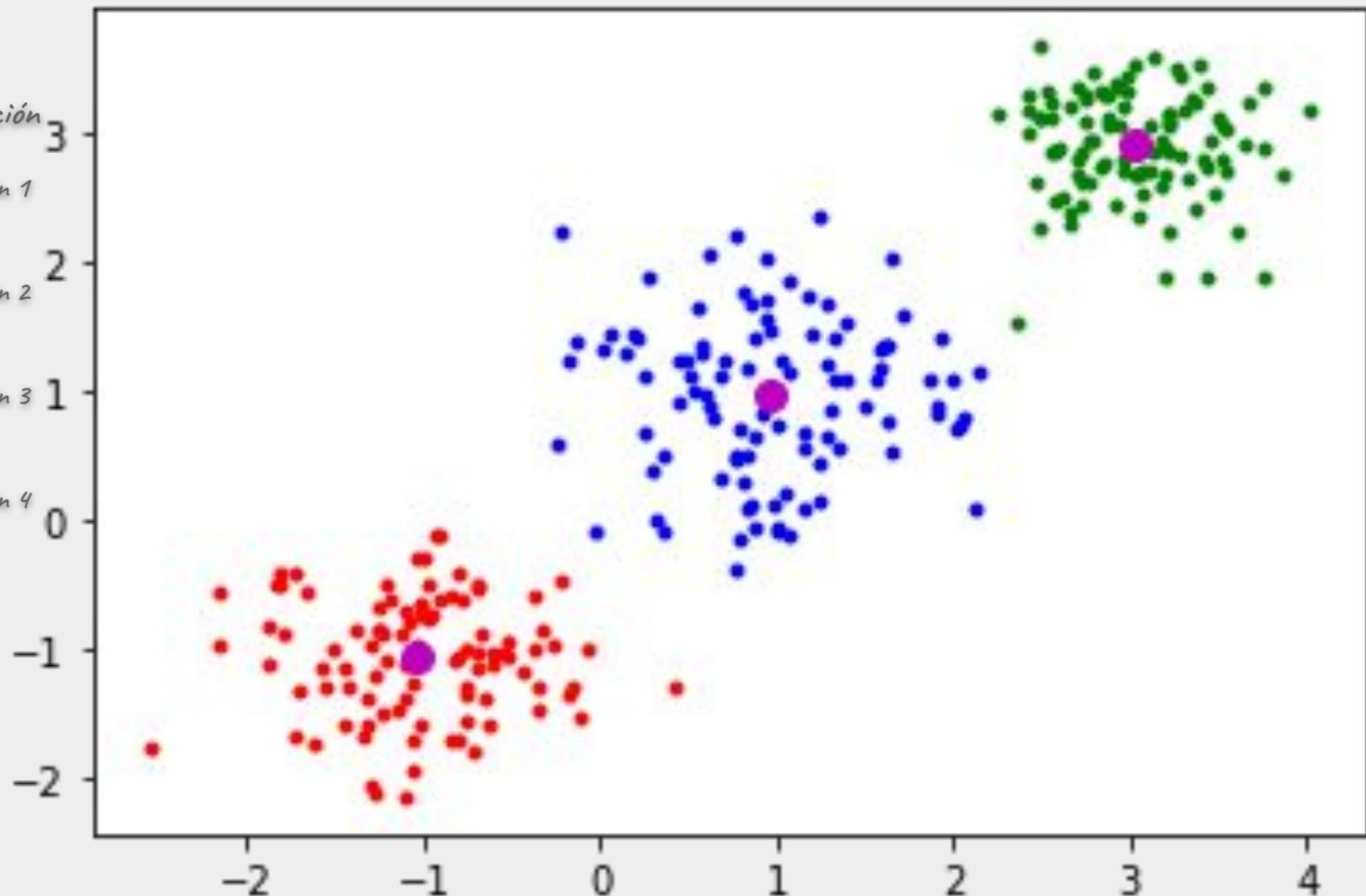
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización } Iteración 4
3. Asignación



# Reubicamos centroides

## Ejecución (Dataset)

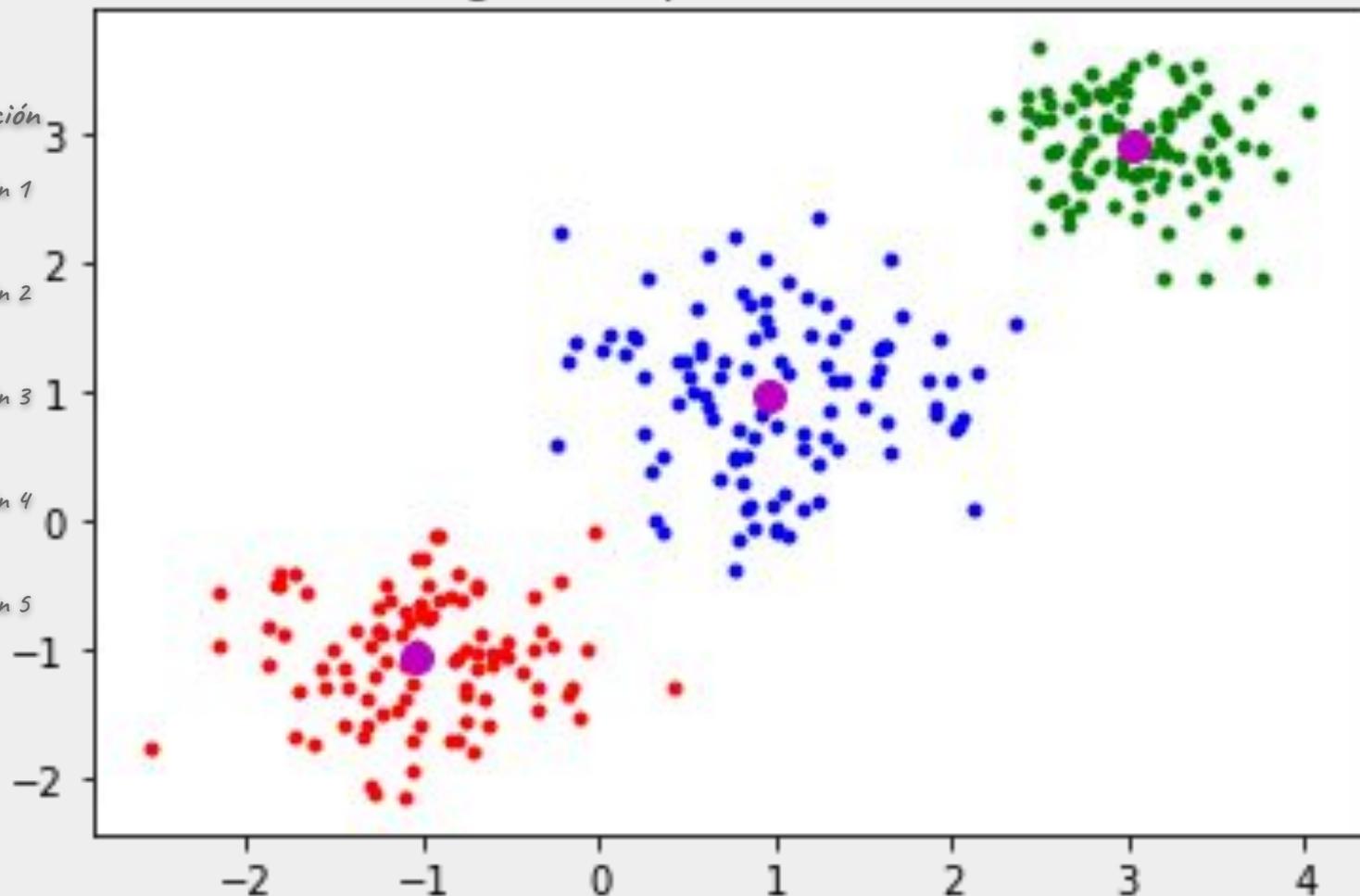
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización } Iteración 4
3. Asignación
2. Actualización



# Reasignamos puntos a centroides

## Ejecución (Dataset)

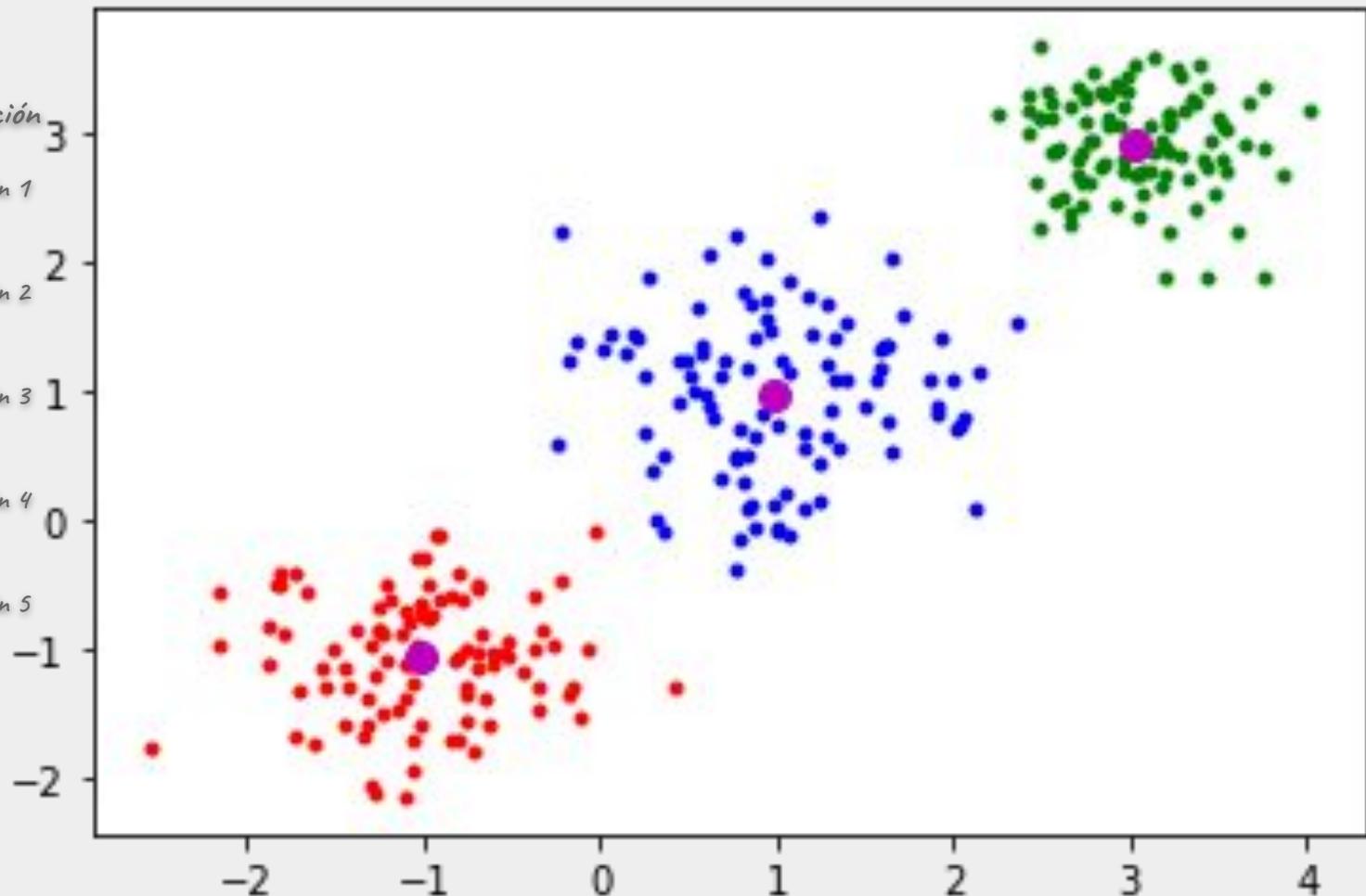
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización } Iteración 4
3. Asignación
2. Actualización } Iteración 5
3. Asignación



# Reubicamos centroides

## Ejecución (Dataset)

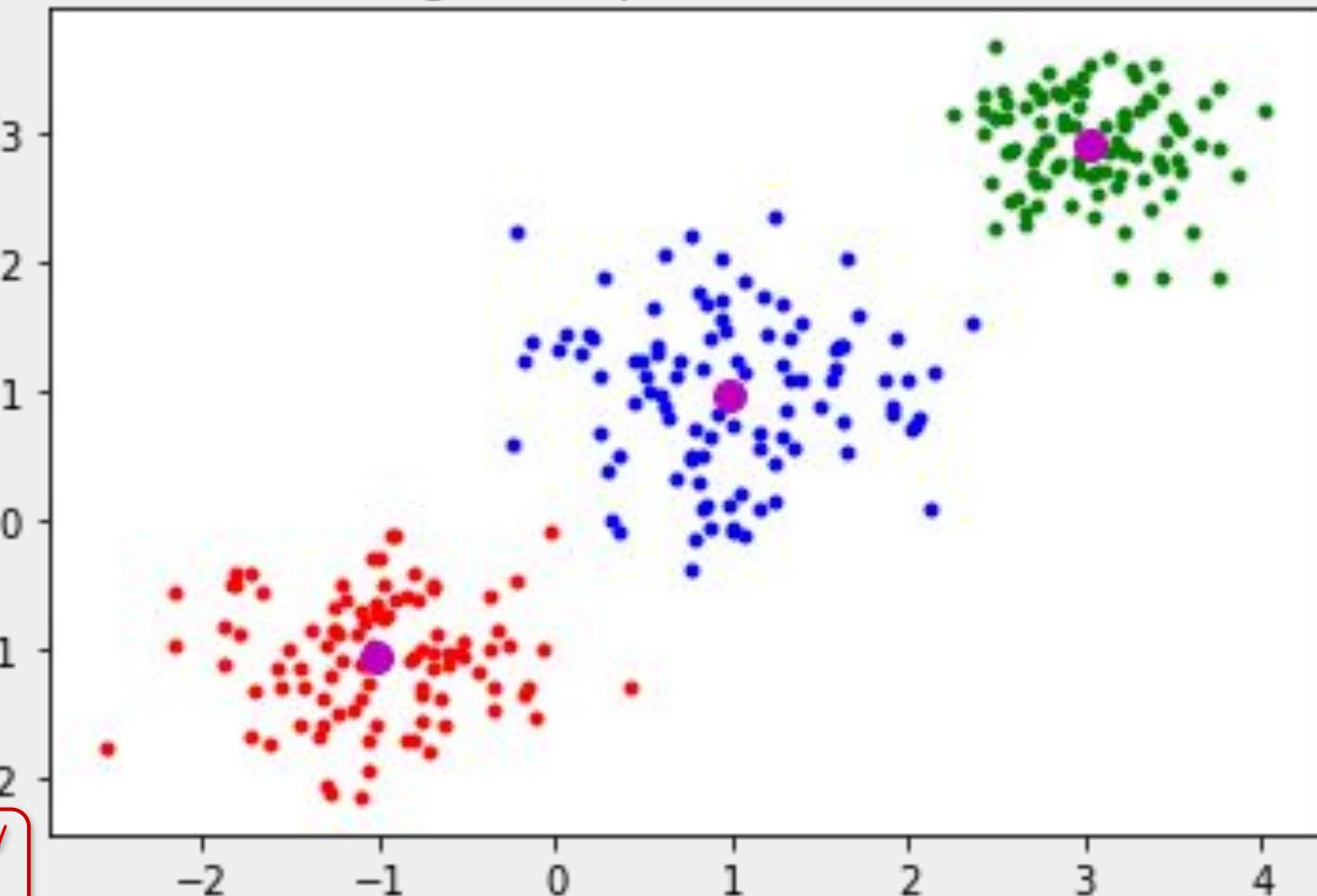
1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización } Iteración 4
3. Asignación
2. Actualización } Iteración 5
3. Asignación
2. Actualización



# Reasignamos puntos a centroides

## Ejecución (Dataset)

1. Inicialización y Asignación
2. Actualización } Iteración 1
3. Asignación
2. Actualización } Iteración 2
3. Asignación
2. Actualización } Iteración 3
3. Asignación
2. Actualización } Iteración 4
3. Asignación
2. Actualización } Iteración 5
3. Asignación
2. Actualización } Iteración 6
3. Asignación



Decidimos cortar por cantidad  
de iteraciones

## Algoritmo K-Medias (K-Means)

Optimización, función de costo a minimizar. Función de **distorsión**.

$$J = \sum_{i=1}^m \sum_{k=1}^K a_{ik} \cdot \|x^{(i)} - \mu_k\|^2$$

K: n° de clusters, m: cant. datos

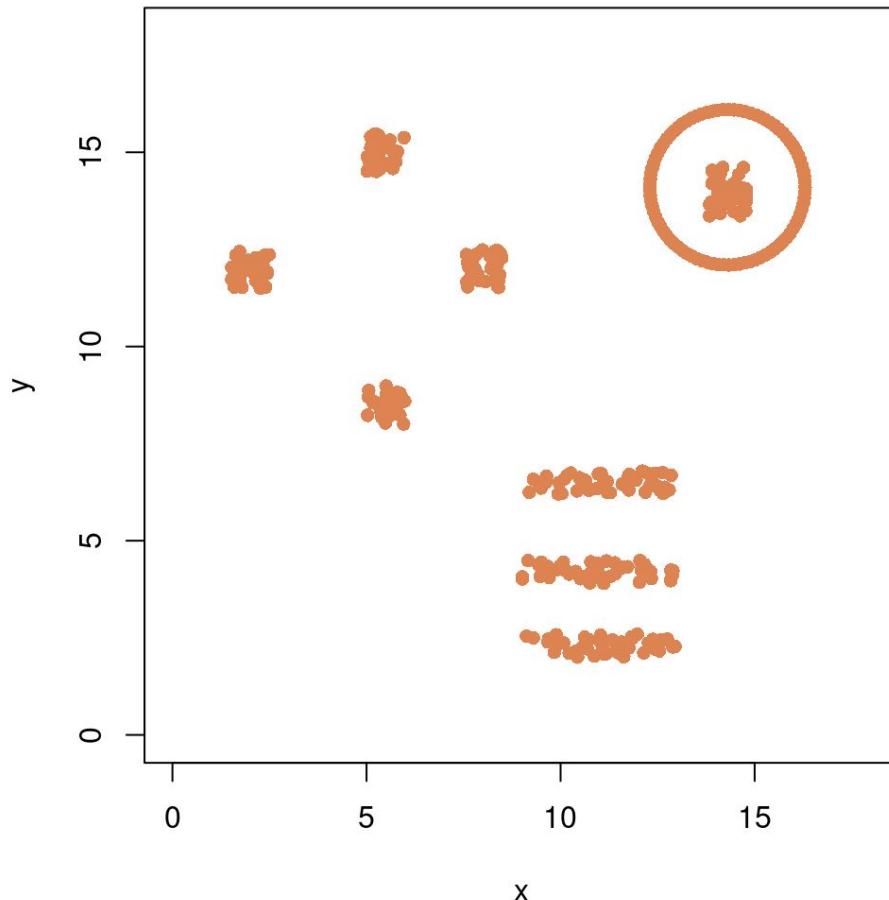
$\mu_k$ : centroide de cluster k

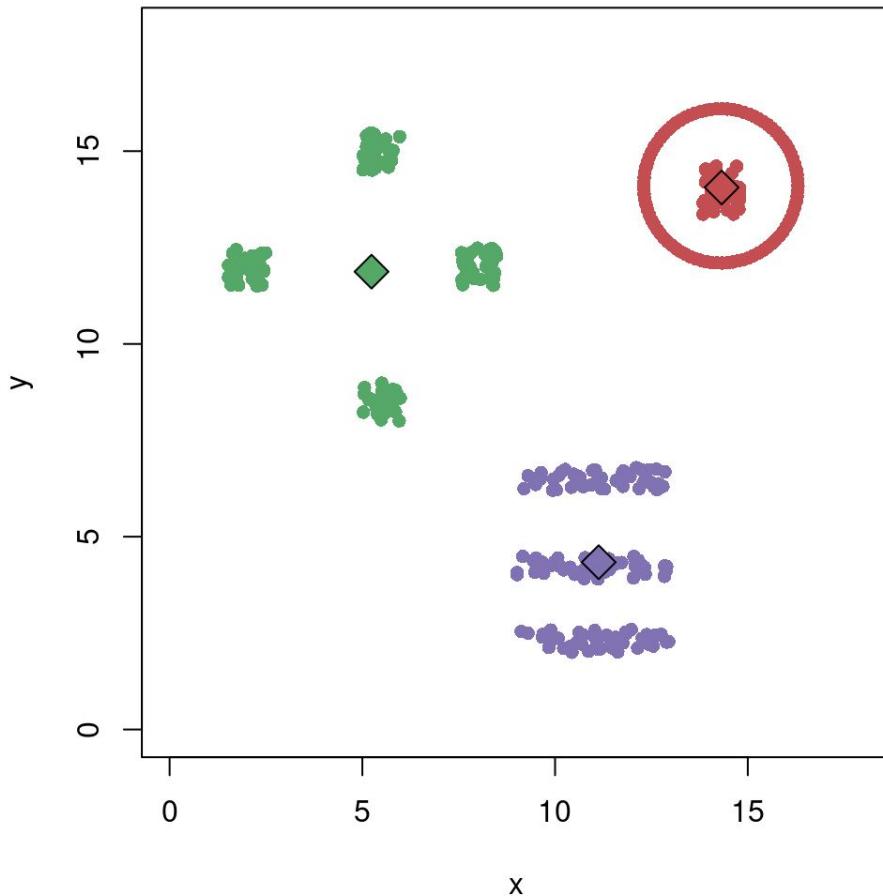
$x^{(i)}$ : dato nro i

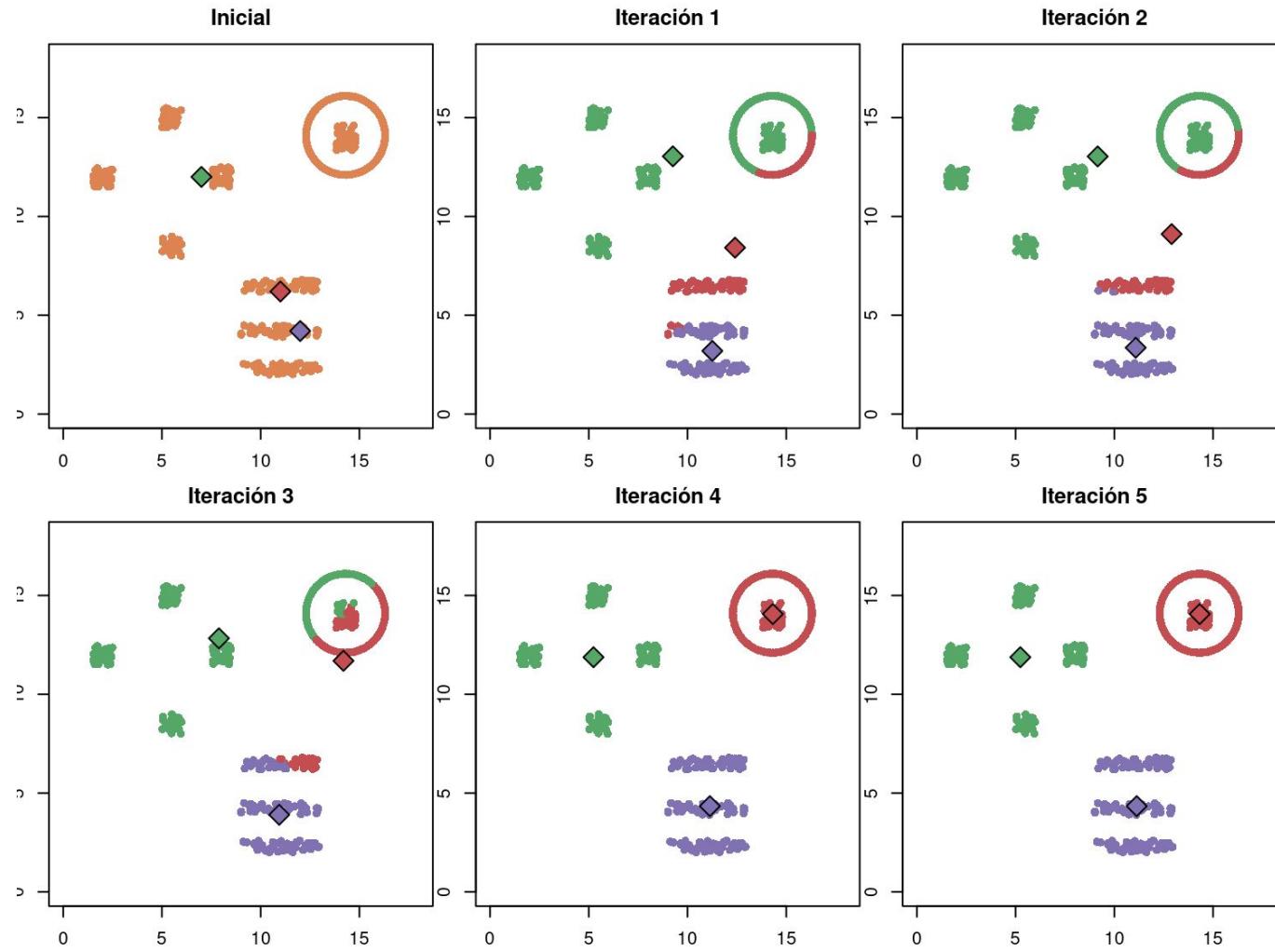
$a_{ik}$   $\begin{cases} 1 & \text{si } x^{(i)} \text{ está asignado al cluster k} \\ 0 & \text{en otro caso} \end{cases}$

K-means intenta encontrar  $\mu_k$  y  $a_{ik}$  que minimicen J

- En **asignación de cluster**: minimiza J con respecto a  $a_{ik}$  (asignando los puntos al centroide más cercano, que está fijo)
- En **actualización de centroide**: minimiza J con respecto a  $\mu_k$

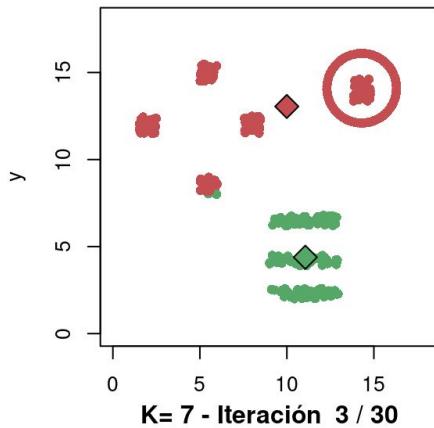




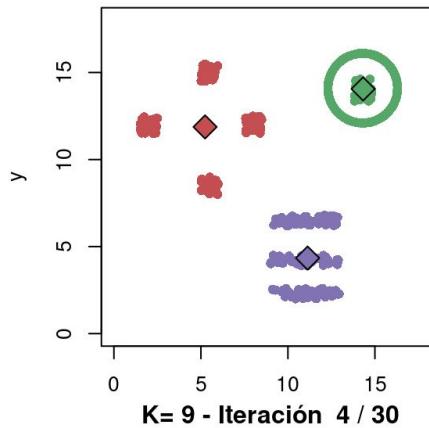


*¿Qué sucede si lo ejecutamos con otros valores de k?*

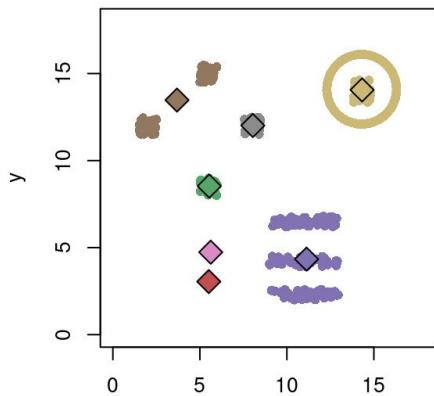
K= 2 - Iteración 4 / 30



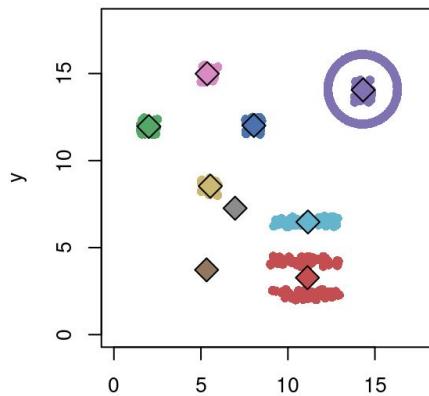
K= 3 - Iteración 6 / 30



K= 7 - Iteración 3 / 30



K= 9 - Iteración 4 / 30



## *¿Cómo elegimos el valor de k?*

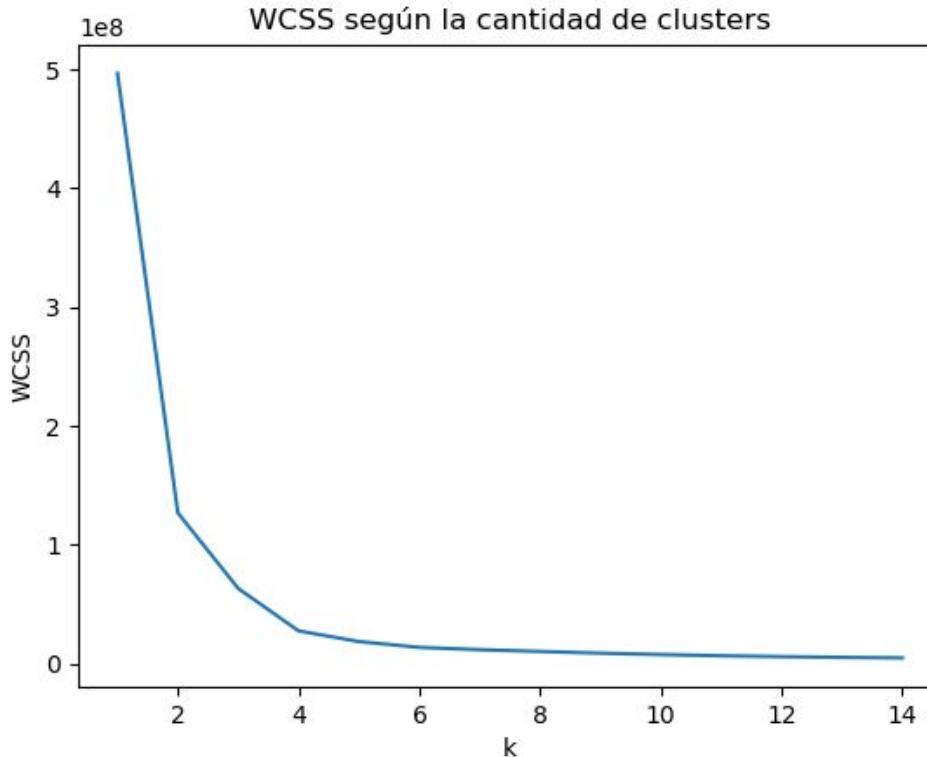
- Una vez agrupados los elementos deberíamos cuantificar cuán bueno (o malo) es el resultado final
- Pero ... ¿Cómo sería una métrica para evaluar si un agrupamiento es bueno (o malo)?
- Proponemos la distancia de cada punto al centro que le asignamos  
Se denomina *Within-Cluster Sum of Squares (WCSS)*:

$$WCSS = \sum_{c_i \in C} \sum_{p_{i,j} \in C_i} \text{distancia}(p_{i,j} - c_i)^2$$

*Siendo  $C$  el conjunto de los centros y  $C_i$  el conjunto de puntos del cluster  $i$*

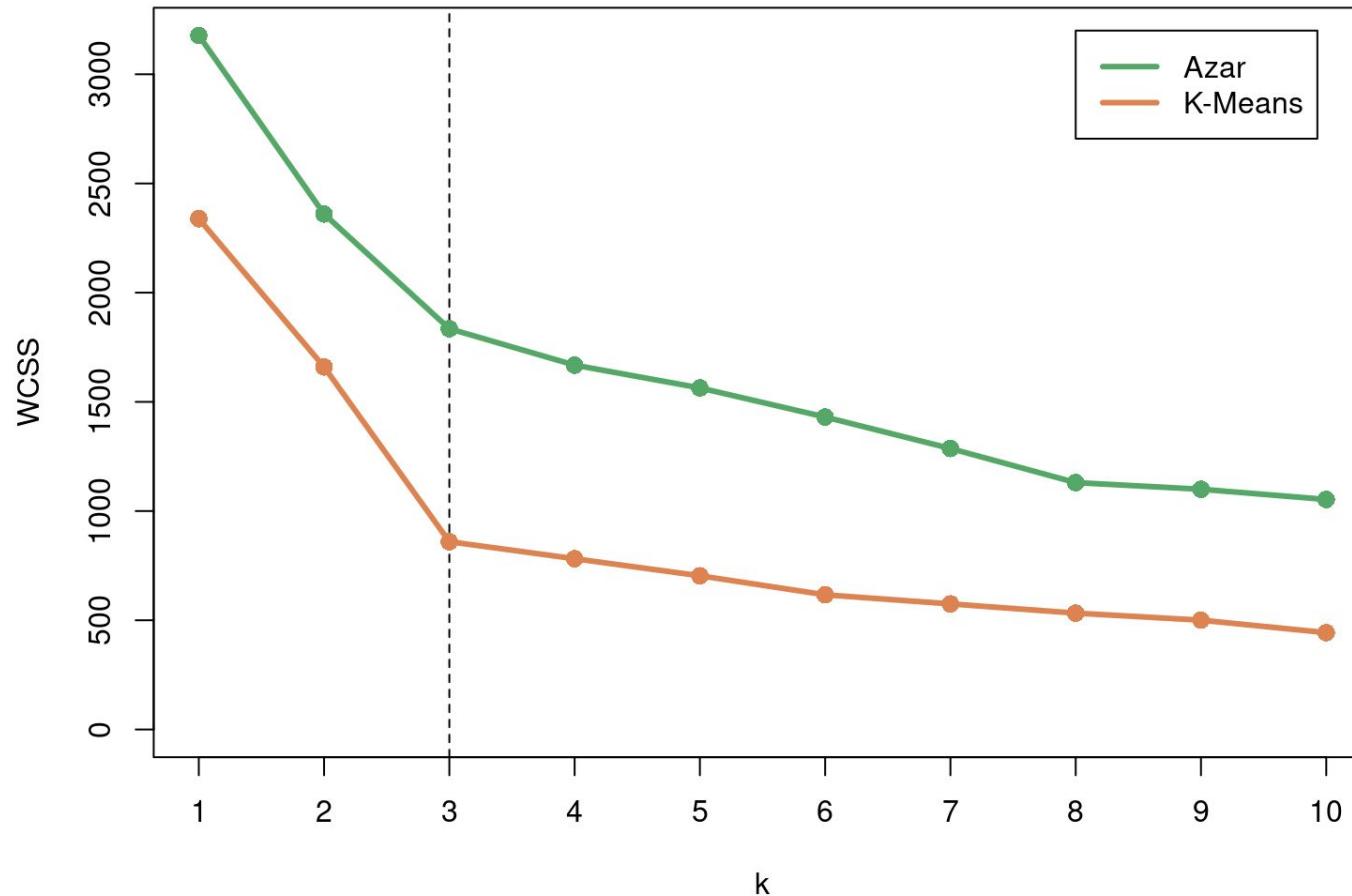
*¿Cómo elegimos el valor de k?*

*Graficamos WCSS para un rango de k y usamos el método del codo*

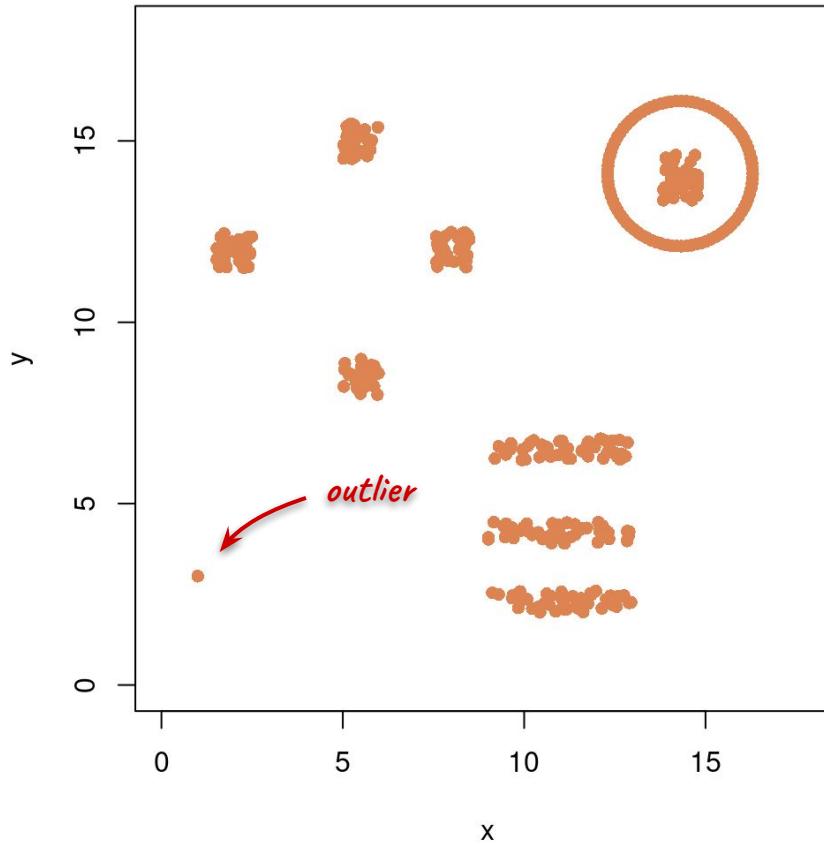


*Pero dado un k, el agrupamiento puede depender de los centroides elegidos al momento de inicializar ...*

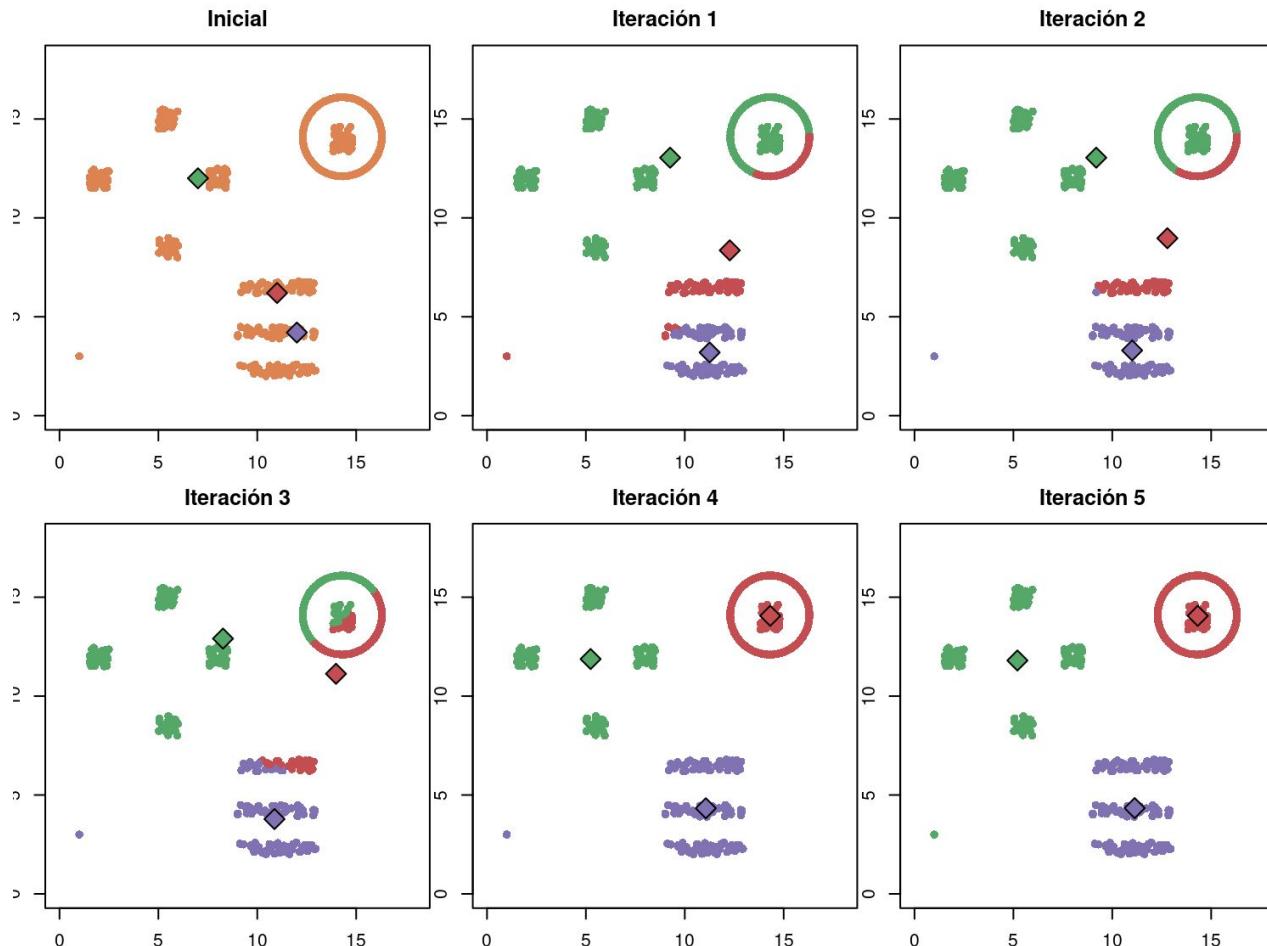
## Promedio de 50 repeticiones para cada k



# ¿Qué sucede con los outliers?



# ¿Qué sucede con los outliers?



## *K-Means con scikit-learn*



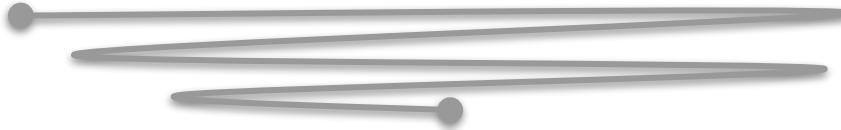
## Algoritmo K-Medias (K-Means)



### Características.

- Rápido y fácil de implementar
- Es necesario especificarle el número de clusters de antemano
- Es sensible a los valores iniciales de los centroides
- No es bueno cuando los datos tienen formas raras, o cuando hay mucha variabilidad en los tamaños de los clusters

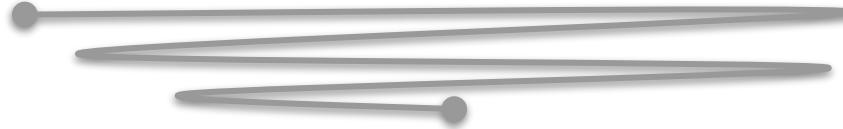
# Aprendizaje No Supervisado - Clustering



*DBSCAN*

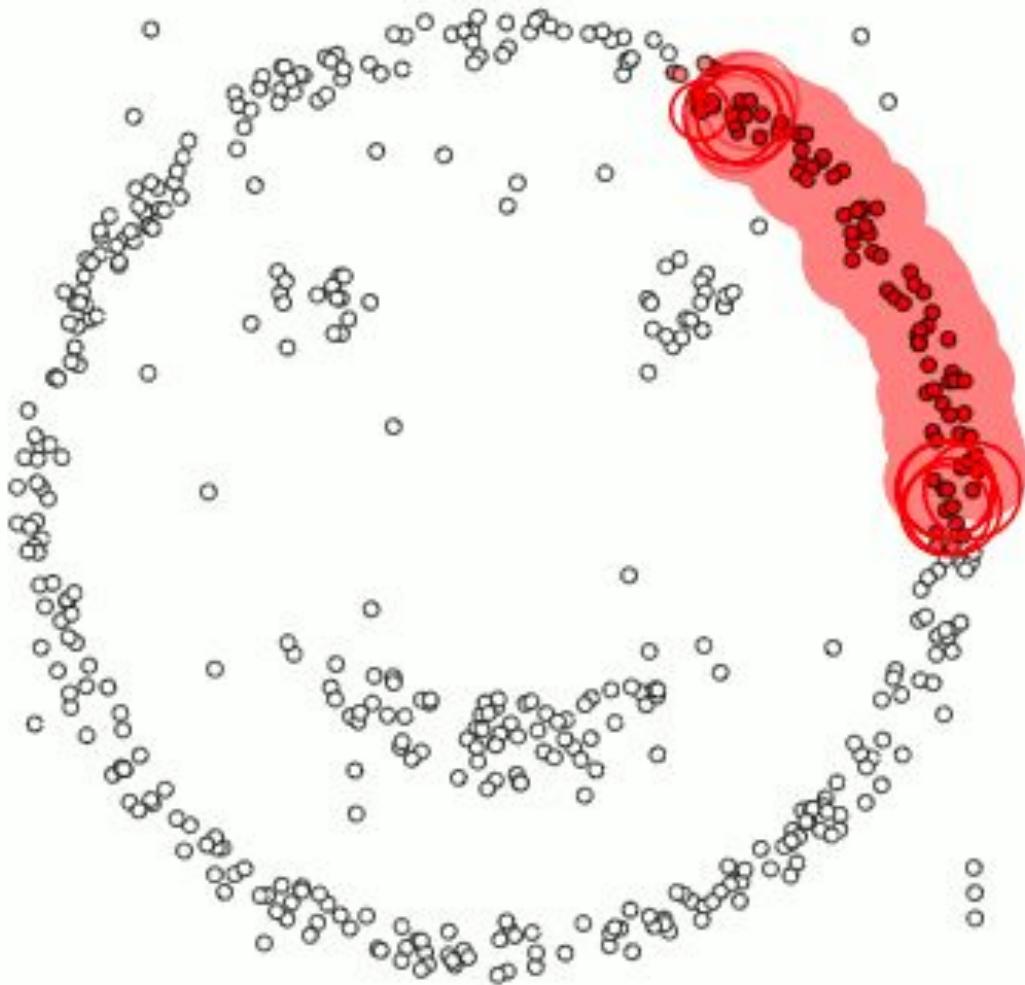
*(Density-Based Spatial Clustering of Applications with Noise)*

# Algoritmo DBSCAN

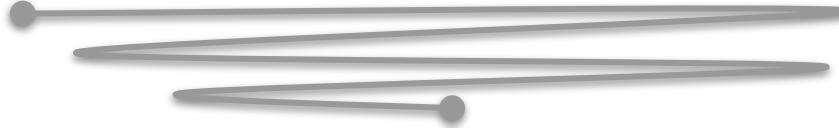


## Algoritmo.

- Un punto  $p$  es un punto núcleo si hay al menos una cantidad  $\text{minPts}$  puntos a una distancia menor a  $\epsilon$  de él
- Un punto  $q$  es alcanzable desde  $p$  si hay un camino de puntos alcanzables que va de  $p$  a  $q$
- Un punto que no sea alcanzable desde al menos  $\text{minPts}$  es considerado ruido



# Algoritmo DBSCAN

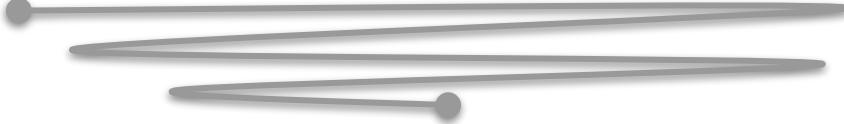


## Algoritmo -> Parámetros

- Cantidad de vecinos requeridos ( $\text{minPts}$ )
- Distancia para la vecindad ( $\epsilon$ )

Importante. NO hay que decidir la cantidad de clusters

## Algoritmo DBSCAN

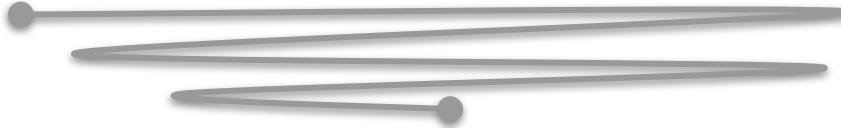


- Para cada observación miramos el número de puntos a una distancia máxima  $\epsilon$  de ella. Esta zona se denomina  $\epsilon$ -vecindad de la observación.
- Si una observación tiene al menos un cierto número de vecinos, incluida ella misma, se considera una observación central. En este caso, se ha detectado una observación de alta densidad.
- Todas las observaciones en la vecindad de una observación central pertenecen al mismo cluster. Puede haber observaciones centrales cercanas entre sí. Por lo tanto, de un paso a otro, se obtiene una larga secuencia de observaciones centrales que constituyen un único cluster.
- Cualquier observación que no sea una observación central y que no tenga ninguna observación central en su vecindad se considera una anomalía/outlier.

*DBSCAN con scikit-learn*



# Algoritmo DBSCAN



## Características.

- Requiere sólo dos parámetros
- No es susceptible al orden en que se encuentran los puntos dentro de la base de datos
- Es robusto detectando outliers
- Puede encontrar clusters con formas geométricas arbitrarias.
- Depende de la noción de distancia
- No puede agrupar bien conjuntos de datos con grandes diferencias en las densidades

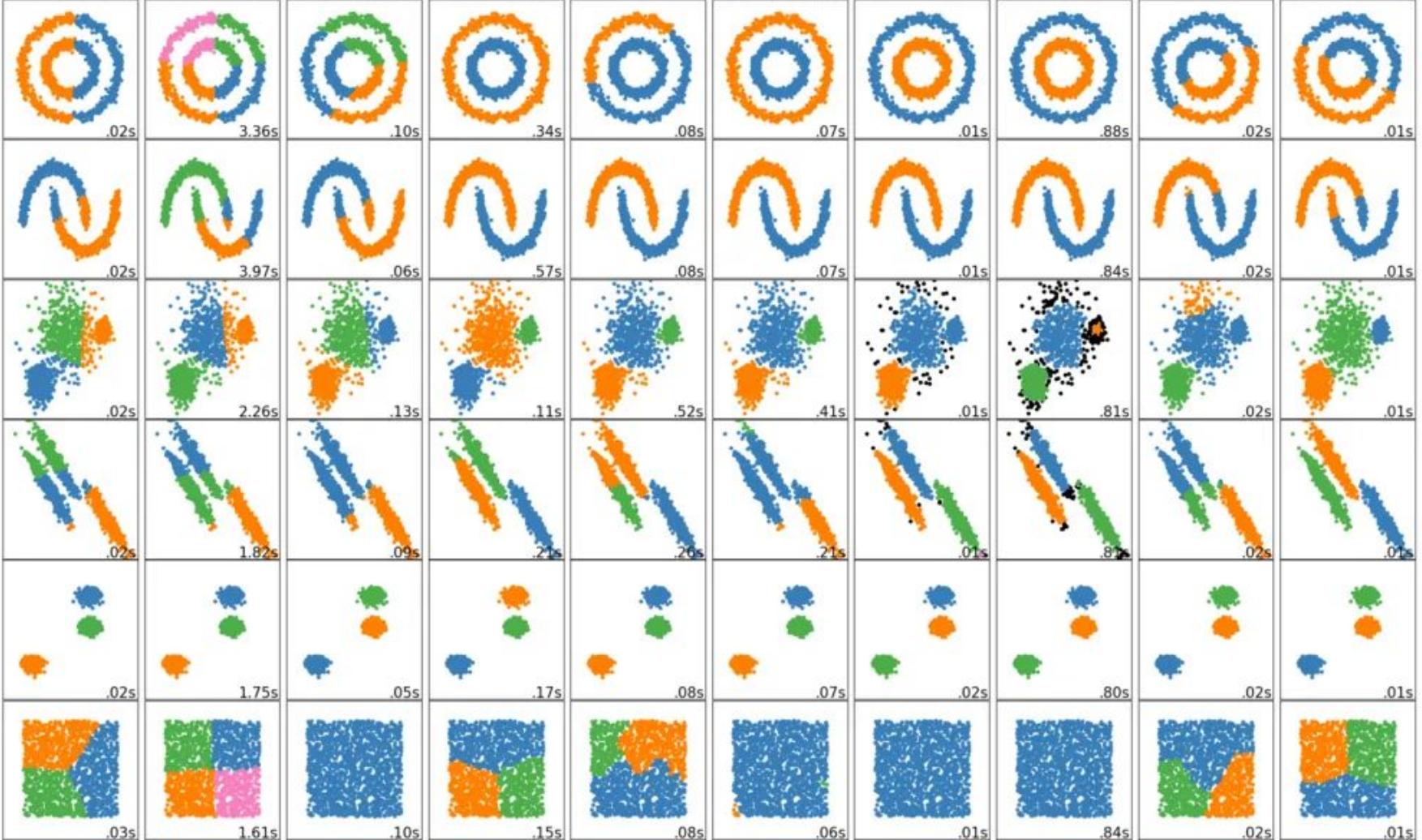
DBSCAN



k-means



MiniBatchKMeans AffinityPropagation MeanShift SpectralClustering Ward AgglomerativeClustering DBSCAN OPTICS Birch GaussianMixture



Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes	Distances between nearest points
OPTICS	minimum cluster membership	Very large n_samples, large n_clusters	Non-flat geometry, uneven cluster sizes, variable cluster density	Distances between points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large n_clusters and n_samples	Large dataset, outlier removal, data reduction.	Euclidean distance between points

*Cierre*



1. *Aprendizaje No Supervisado*
2. *KMeans*
3. *DBSCAN*